

Grado en Ingeniería Electrónica Industrial y Automática
2018-2019

Trabajo Fin de Grado

“Orientación relativa del vehículo en el carril para coches autónomos”

“EGO-Lane localization for autonomous driving”

Jorge Velado Núñez

Tutor

Francisco Miguel Moreno Olivo

Lugar y fecha de presentación prevista

08 de julio de 2019, 9:30 en el aula 7.2.J.04 de la EPS UC3M



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

RESUMEN

Uno de los mayores retos automovilísticos actuales es el desarrollo de una conducción autónoma completa. Existen numerosos grupos de investigación y grandes compañías trabajando en ello y los avances en las últimas cuatro décadas han permitido llegar a donde estamos hoy, donde el coche autónomo es casi una realidad.

Nowadays, one of the biggest automotive goals is to develop a full driverless car. A high number of researchers and companies are working on a prototype for being successful in this challenge and the autonomous driving is almost a reality.

En este proyecto de fin de grado, en relación con la conducción autónoma e integrado en el trabajo conjunto realizado por el Laboratorio de Sistemas Inteligentes de la Universidad Carlos III de Madrid, se obtienen las tres variables necesarias para el control lateral del automóvil y la localización de este en el carril: la desviación angular de la trayectoria del vehículo respecto a la trayectoria de la carretera y el desplazamiento lateral izquierdo y derecho, es decir, la distancia existente entre el vehículo y las líneas que delimitan el carril por ambos lados, siendo medida perpendicularmente a la trayectoria de la carretera.

In this Bachelor Thesis, related with the autonomous driving and working inside the research group of the Intelligent Systems Laboratory at the University Carlos III de Madrid, the three variables needed for the vehicle lateral control and the EGO-lane localization are obtained: angle displacement from the road trajectory and the right and left lateral displacement. These last two variables are taken perpendicular to the road trajectory.

Para ello se procesan y analizan los datos de las imágenes de video obtenidas por una cámara instalada en la parte frontal del vehículo tras realizar una detección y delimitación del carril en las mismas. También se proponen y estudian dos curvas diferentes como modelos de carretera, parábola y clotoide, para obtener las variables mencionadas anteriormente: desviación angular y lateral del vehículo.

For this project, the data taken from a camera installed in the front part of the car is analyzed, after having been processed to detected and delimited the road lane on the images. Two different kind of road models, parabola and clotoide, are also proposed in order to obtain the variables mentioned above: angular and lateral displacement of the vehicle on the road.

Palabras clave

Conducción autónoma, localización, carril, carretera, desviación angular, distancia lateral, LKS, control, vehículos inteligentes

Keywords

Autonomous driving, driverless, EGO-lane, localization, road, angular, lateral, displacement, LKS, control, intelligent vehicles

DEDICATORIA

Me gustaría hacer mención especial y en primera instancia a Luis Miguel Velado Rosado y a Mercedes Núñez Municio, mis padres. Ellos me han enseñado y llevado a donde estoy hoy. Cualquier mérito que alcance habrá sido más suyo que mío.

A mis abuelos, por su fe, esfuerzo y apoyo incansable, su alegría por verme crecer y por ser una parte de mí tan grande.

A Anaëlle, porque dos semanas y media pasan muy rápido, y no hay mejor persona que ella. Sin duda, lo mejor de todos estos años.

A todos mis compañeros de la universidad, y a mis amigos de siempre.

A mi tutor, Fran, por la oportunidad de aprender y su ayuda en este proyecto.

Y a todos aquellos que han estado conmigo en el día a día, y lo seguirán estando.

GRACIAS.

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN.....	1
1.1. ADAS y LKS.....	2
1.2. Objetivos y motivación.....	4
1.3. Estado del arte.....	5
2. ANÁLISIS Y DISEÑO DEL SISTEMA.....	10
2.1. Recursos iniciales.....	10
2.2. Entorno de simulación.....	11
2.2.1. ROS.....	11
2.2.2. OpenCV.....	12
2.3. Implementación.....	13
2.3.1. Obtención de los datos.....	14
2.3.2. Desarrollo modelo de carretera.....	22
2.3.2.1. Modelo cuadrático o parabólico.....	23
2.3.2.2. Modelo cúbico o clotoidal.....	23
2.3.2.3. Obtención parámetros de la carretera.....	24
2.3.3. Obtención punto de la curva.....	26
2.3.3.1. Modelo cuadrático o parabólico.....	27
2.3.3.2. Modelo cúbico o clotoidal.....	29
2.3.3.3. Soluciones al modelo de carretera.....	32
2.3.4. Desviación angular.....	32
2.3.5. Desplazamiento lateral.....	35
2.3.6. Visualización.....	37
3. RESULTADOS	40
4. PLANIFICACIÓN.....	45
5. MARCO REGULADOR.....	47
6. ESTUDIO SOCIO-ECONÓMICO.....	50
6.1. Impacto socio-económico.....	50
6.2. Presupuesto.....	50
7. CONCLUSIÓN.....	54
7.1. Líneas de trabajo futuro.....	54
BIBLIOGRAFIA.....	57
ANEXO. ESPECIFICACIONES TÉCNICAS CÁMARA BASLER	

ÍNDICE DE FIGURAS

FIG. 1. DEFINICIÓN GRÁFICA LKS [27]	4
FIG. 2. ESQUEMA ALGORITMO ADAPTATIVO [29]	6
FIG. 3. EJEMPLO IPM E IPM ADAPTATIVA [30]	7
FIG. 4. ESQUEMA CÁLCULO ANALÍTICO DE LA DESVIACIÓN ANGULAR Y LATERAL [38]	8
FIG. 5. IMAGEN DE ADA	10
FIG. 6. ESQUEMA ORGANIZACIÓN DEL SISTEMA	13
FIG. 7. IMAGEN DE LA CARRETERA NUMERADA	15
FIG. 8. EJEMPLO CONVERSIÓN MÁSCARA BLANCO Y NEGRO	16
FIG. 9. EJEMPLO DILATACIÓN DE LA MÁSCARA	17
FIG. 10. EJEMPLO EROSIÓN DE LA MÁSCARA	18
FIG. 11. EJEMPLO MÁSCARA FINAL	19
FIG. 12. EJEMPLO DETECCIÓN CONTORNO DEL CARRIL	19
FIG. 13. EJEMPLO OBTENCIÓN DATOS DEL CARRIL	20
FIG. 14. EJEMPLO DE CLOTOIDE [46]	24
FIG. 15. EJEMPLO MATRIZ SISTEMA PARABÓLICO	25
FIG. 16. EJEMPLO MATRIZ MODELO CLOTOIDIAL	25
FIG. 17. EJEMPLO MATRIZ RESUELTA MODELO PARABÓLICO	25
FIG. 18. EJEMPLO MATRIZ RESUELTA MODELO CLOTOIDIAL	26
FIG. 19. EJEMPLO PARÁMETROS DE LA CARRETERA, MODELO PARABÓLICO	26
FIG. 20. EJEMPLO PARÁMETROS DE LA CARRETERA, MODELO CLOTOIDIAL	26
FIG. 21. EJEMPLO IMAGEN DE LA CARRETERA	28
FIG. 22. EJEMPLO RESULTADOS ECUACIÓN MODELO PARABÓLICO	29
FIG. 23. EJEMPLO RESULTADOS ECUACIÓN MODELO CLOTOIDIAL	31
FIG. 24. ESQUEMA CALCULO DESVIACIÓN ANGULAR	34
FIG. 25. ESQUEMA CÁLCULO DESPLAZAMIENTO LATERAL	36
FIG. 26. EJEMPLO DE VISUALIZACIÓN DE RESULTADOS EN LA PANTALLA	38
FIG. 27. RESULTADO IMAGEN 1 MODELO PARABÓLICO	40
FIG. 28. RESULTADO IMAGEN 1 MODELO CLOTOIDIAL	40
FIG. 29. RESULTADO IMAGEN 2 MODELO PARABÓLICO	41
FIG. 30. RESULTADO IMAGEN 2 MODELO CLOTOIDIAL	41
FIG. 31. RESULTADO IMAGEN 3 MODELO PARABÓLICO	41
FIG. 32. RESULTADO IMAGEN 3 MODELO CLOTOIDIAL	41
FIG. 33. RESULTADO IMAGEN 4 MODELO PARABÓLICO	41
FIG. 34. RESULTADO IMAGEN 4 MODELO CLOTOIDIAL	41
FIG. 35. RESULTADO IMAGEN 5 MODELO PARABÓLICO	42
FIG. 36. RESULTADO IMAGEN 5 MODELO CLOTOIDIAL	42
FIG. 37. RESULTADO IMAGEN 6 MODELO PARABÓLICO	42
FIG. 38. RESULTADO IMAGEN 6 MODELO CLOTOIDIAL	42
FIG. 39. DIAGRAMA DE GANTT DE LA PLANIFICACIÓN	45

ÍNDICE DE TABLAS

TABLA I RESULTADOS	43
TABLA II COSTES DE PERSONAL.....	51
TABLA III COSTES MATERIALES	51
TABLA IV RESUMEN PRESUPUESTO	52

1. INTRODUCCIÓN

La continua evolución de la tecnología y de las condiciones de vida del ser humano crecen exponencialmente. Los medios de transporte, cada día más accesibles y universales, permiten acercar lugares, productos y personas, convirtiéndose en uno de los puntos clave de nuestra economía y sociedad.

Tras la aparición del automóvil a finales del siglo XIX y su popularización en las primeras décadas del siglo XX [1], éste se convirtió rápidamente en el principal método de transporte para una inmensa mayoría de la población. La idea de un coche sin conductor no tarda en llegar y en 1925, ya se asistió a la primera demostración en público por parte de Houdina Radio Control Co. en las calles de Nueva York mediante radiocontrol [2].

En la Exposición Universal de 1939 [3], también en la ciudad de Nueva York, Norman Bel Geddes presentó una exhibición llamada “Futurama” en la que, mediante una maqueta, rediseñaba las ciudades tal y como se conocían en aquella época. Una de las principales novedades fue la idea de coches eléctricos autónomos monitorizados por radio que circulaban por carreteras automáticas, capaces de comunicar diferentes ciudades de EE. UU. alcanzando velocidades de 160 km/h.

No fue hasta 1986, cuando Ernst Dickmanns, considerado un pionero de los automóviles sin conductor, creó el primer coche capaz de circular por si solo por la Universidad de Múnich. En 1994, con una furgoneta Mercedes 500 SEL, equipada con cámaras, sensores, y sistemas de visión artificial, consiguió recorrer finalmente 1000 km en la circunvalación de París. Un año más tarde, Dickmanns puso a punto un coche Mercedes Clase S, permitiéndole alcanzar 180 km/h y realizar el trayecto Múnich-Dinamarca ida y vuelta, 1678 km de autopista [4].

A principios de este siglo y con la entrada del nuevo milenio, surgieron numerosos nuevos proyectos y equipos de investigación relacionados con este tema. Y a su vez nuevos retos, carreras y competiciones de automóviles sin conductor. En 2004, se lanza por parte de DARPA (Defense Advanced Research Projects Agency), el gran reto de completar el trayecto en EE. UU. entre Barstow, California y Primm, Nevada. Este reto fue completado en su segunda edición por el Standford Racing Team, galardonado con 2 millones de dólares [5]. En los últimos años han seguido surgiendo otras grandes competiciones como Roborace [6] o Driverless Formula Student [7].

Las últimas noticias de coches autónomos vienen de la mano de grandes compañías como Uber, Tesla o Waymo, filial de Google. Uber y Waymo ya cuentan con una flota de taxis autónomos (aunque todavía llevan un conductor para posibles emergencias) [8] [9] y, además, Waymo anunció el pasado enero la construcción de la primera planta dedicada exclusivamente a la construcción de coches autónomos en masa [10]. Por su parte, Tesla cuenta con la recién estrenada tecnología autopilot [11], que prevé poner en funcionamiento en las calles para 2020.

Gracias a este largo recorrido de hitos tecnológicos, la conducción autónoma permite encontrar soluciones a los problemas actuales y futuros que producen y producirán los medios de transporte, concretamente, en este caso, a los problemas generados por los automóviles.

Los datos más recientes, publicados en 2016 por la Organización Mundial de la Salud (OMS), muestran los accidentes de tráfico como una de las 10 principales causas de muerte, subiendo dos puestos en relación con estudios realizados en el año 2000. Se estima que 1,4 millones de personas perdieron la vida por este motivo en 2016 [12], convirtiéndose en la principal causa de mortalidad en niños y jóvenes (entre 5 y 29 años) [13]. Solo en España hubo 1.180 víctimas mortales el año pasado [14]. Entre los principales factores de riesgo, podemos encontrar el exceso de velocidad, la conducción bajo los efectos del alcohol y otras drogas y la conducción distraída [13], factores relacionados con el error humano que se podrían evitar. Según publicaciones del International Transport Forum (ITF), se reduciría un 20% el número de siniestros si no se excediera el límite obligatorio de velocidad [15].

Un estudio reciente publicado por INRIX Global Traffic Scorecard revela otro de los grandes problemas de la conducción. La congestión del tráfico en Reino Unido costó alrededor de 8 billones de libras en 2018 [16] y un ciudadano americano pierde 97 horas al año de media por este motivo, lo que se traduce en 1,348 dólares al año (contando tiempo, combustible y valor de emisiones de carbono generadas) [17]. Ciudades como Londres, que se sitúa dentro de las diez ciudades con más tráfico del mundo, perdieron más de 200 horas al año por conductor. Madrid es la ciudad española con más tráfico, seguida de Barcelona y Valencia, situadas en los puestos 22, 38 y 53 respectivamente [18].

Por otra parte, la preocupación y concienciación por la conservación del medio ambiente es cada vez mayor y no se pueden obviar datos como que los medios de transporte emiten más del 30% de los gases contaminantes en Europa. El 72% de estos gases son producidos por el transporte en carretera. Al contrario que otros sectores como la industria, energía o agricultura, que han tendido a una reducción progresiva de sus emisiones de CO₂, el sector del transporte sigue una tendencia al alza [19]. Por ello, cada vez son más frecuentes las noticias relacionadas con medidas regulatorias en cuanto a la emisión de gases contaminantes [20] [21].

Otro inconveniente actual es la imposibilidad de un acceso universal a la conducción, es decir, para la conducción son necesarias unas capacidades y aptitudes que excluyen a ciertos grupos de la sociedad como personas de la tercera edad y/o con ciertas discapacidades.

1.1. ADAS y LKS

A día de hoy, como ya se ha visto, existen vehículos totalmente autónomos capaces de realizar grandes trayectos sin la ayuda de ningún conductor. Dependiendo del grado de autonomía del vehículo y de las tareas encomendadas al conductor se pueden definir seis niveles de vehículos autónomos.

Así define estos niveles la Dirección General de Tráfico (DGT) [22]:

- Nivel 0, sin automatización. El conductor realiza continuamente todas las tareas asociadas a la conducción, incluso cuando son mejoradas a través de algún aviso o la intervención de sistemas.
- Nivel 1, conducción asistida. El sistema de ayuda a la conducción desarrolla una tarea específica, bien realiza la conducción dinámica lateral o longitudinal utilizando la información del entorno del vehículo, mientras que el conductor realiza el resto de tareas de conducción.
- Nivel 2, conducción parcialmente automatizada. El sistema de ayuda a la conducción desarrolla la conducción dinámica lateral y longitudinal utilizando la información del entorno del vehículo, mientras que el conductor realiza el resto de tareas de la conducción.
- Nivel 3, conducción automatizada condicionada. El sistema de conducción automatizada desarrolla todas las tareas de la conducción con la expectativa de que el conductor responda adecuadamente a la petición de intervención por parte de éste.
- Nivel 4, conducción altamente automatizada. El sistema de conducción automatizada desarrolla todas las tareas de la conducción, incluso si el conductor no responde adecuadamente a la petición de intervención por parte de éste.
- Nivel 5, conducción plenamente automatizada. El sistema de conducción automatizada desarrolla todas las tareas de la conducción bajo todas las circunstancias de la vía y ambientales.

Sin embargo, la tendencia actual, debido a la necesidad de mejora de estos prototipos, los problemas legales, éticos [23] y errores graves cometidos con fatales consecuencias [24], es integrar en los automóviles sistemas avanzados de ayuda a la conducción (ADAS, Advanced Driver Assistance System), dando al conductor todavía un amplio margen de decisión, que se irá reduciendo con el tiempo.

Estos sistemas se basan en la percepción del entorno del automóvil mediante cámaras, sensores y satélites, planificando y llevando a cabo una maniobra acorde a la situación concreta vivida en la carretera que ayude al conductor.

Los sistemas actuales más comunes son la frenada autónoma de emergencia (AEB, Autonomous Emergency Braking), la asistencia al abandono involuntario de carril (LKS, Lane Keeping System), el sistema de alerta de tráfico cruzado, el control de crucero adaptativo, la alerta de ángulo muerto (BSD, Blind Spot detection), el sistema de reconocimiento de señales (TSA, Traffic Sign Alert) y los sistemas de asistencia al aparcamiento (PA, Park Assist) [25].

Dentro de estos sistemas de ayuda a la conducción, la asistencia al abandono involuntario de carril (LKS) está relacionado directamente con el control lateral del vehículo, para el cual se desarrolla este Trabajo de Fin de Grado.

El sistema LKS monitoriza las marcas y posición de la carretera a la vez que provee información de las condiciones del vehículo, teniendo como objetivo incrementar la seguridad y mantener el automóvil en el centro del carril. Para ello, toma el control total del vehículo, si es activado previamente por el conductor, o produce algún tipo de alarma como un ligero movimiento del volante o una alarma acústica (LDWS, lane-departure warning system) [26].

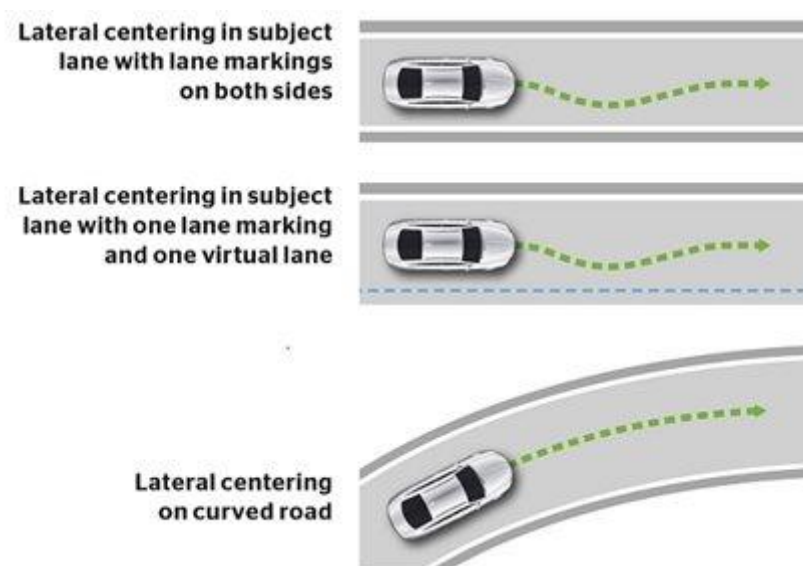


Fig. 1. Definición gráfica LKS [27]

1.2. Objetivos y motivación

En este marco contemporáneo, amparados por la tecnología, continúa y surge con mayor vigor la idea de un vehículo autónomo eléctrico o de bajas emisiones que minimice y evite el error humano y la emisión de gases contaminantes a la atmosfera, favoreciendo a la creación de una red de transporte común más eficiente e inteligente. De esta manera, las decisiones individuales de cada conductor (refiriéndose a conductor a la conducción autónoma) se verían precedidas por el contexto y situación que le rodea, totalmente previsibles y premeditadas. Calculando siempre la ruta más óptima que permita reducir el consumo, el tráfico y el tiempo que se pasa en la carretera.

Uno de los puntos más importantes para llevar a cabo la conducción autónoma es la

situación del vehículo en el entorno, ya que de esta primera percepción dependerán el resto de los sistemas de control del vehículo. Para ello hay que analizar los obstáculos que lo rodean y la vía por la que circula.

Aunque en ciertas ocasiones el automóvil se ve obligado a circular por calzadas sin asfaltar en vías rurales, lo ideal y necesario es una carretera que permita la correcta circulación del automóvil y la seguridad de sus ocupantes. Hoy en día, existe una gran red de carreteras que cumplen esta labor por las que circulan a diario millones de vehículos.

Dentro de los sistemas ADAS y desarrollando la asistencia al abandono involuntario de carril para el control de la conducción autónoma completa, se encuentra como elemento indispensable la localización relativa del vehículo respecto del carril.

Este proyecto está integrado en el trabajo conjunto que se realiza en el laboratorio de sistemas inteligentes de la universidad Carlos III de Madrid en relación con la conducción autónoma.

Para la toma de decisiones y control lateral del vehículo se requieren tres variables, las cuales se pretenden obtener en este Trabajo de Fin de Grado:

- La desviación angular de la trayectoria del vehículo respecto a la trayectoria de la carretera.
- El desplazamiento lateral izquierdo y derecho. La distancia existente entre el vehículo y las líneas que delimitan el carril por ambos lados, siendo medida perpendicularmente a la trayectoria de la carretera.

El proyecto pretende extraer los datos de las imágenes obtenidas por una cámara instalada en la parte frontal del vehículo en las que se ha detectado y delimitado el carril en el que se encuentra previamente. También se analizarán diversos modelos de carretera, estudiando la curvatura que mejor se adapte. De esta manera se pretende reducir el error de la detección del carril y obtener de la manera más fiable posible las tres variables de salida que se requieren para el control del vehículo.

El objetivo del diseño e implementación del sistema será definido para autopistas y carreteras que mantengan una trayectoria recta y/o con una curvatura semi-recta.

1.3. Estado del arte

Existen numerosos estudios en relación con la conducción autónoma y muchas empresas especializadas. Se ha convertido en un negocio y en una carrera tecnológica por parte de

las compañías automovilísticas por lanzar al mercado un prototipo de estas características, lo que supone una restricción de conocimientos compartidos y una complicación añadida al acceso de información.

En este apartado se analizan las diferentes soluciones existentes a día de hoy en relación con la extracción de datos de la carretera y su posterior análisis. No se tiene en cuenta aquellos estudios que usan GNSS (Global Navigation Satellite System), ya que el objetivo de este Trabajo de Fin de Grado es obtener las variables necesarias de una manera que proporcione una fiabilidad extra basada en la detección en tiempo real de nuestro entorno, fiabilidad que el GNSS no puede garantizar debido a la complejidad geométrica de los ambientes urbanos y los errores generados por la pobre recepción de las señales de radio [28].

Para analizar en tiempo real el entorno se utiliza una o varias cámaras colocadas en la parte frontal del vehículo. Con estas imágenes se detectan las líneas que delimitan el carril de la carretera y se procede a la obtención de los parámetros de la carretera: curvatura, anchura del carril, pendiente y número de carriles.

Para ello, se utiliza un algoritmo adaptativo [29] con una calibración dinámica, dividiendo el proceso en cuatro pasos: reconocimiento del carril, conversión de coordenadas, estimación de los parámetros de la carretera y calibración del punto de vista. Ya que el reconocimiento del carril se hace en una imagen plana, se usan una serie de ecuaciones que relacionan las coordenadas del mundo real y las de la imagen obtenida. En este caso [29], utilizan una ecuación cuadrática para definir el modelo de carretera. En la segunda parte del estudio en el que se desarrolla el control para la asistencia en carretera se utiliza la desviación lateral, la curvatura y la velocidad como variables iniciales.

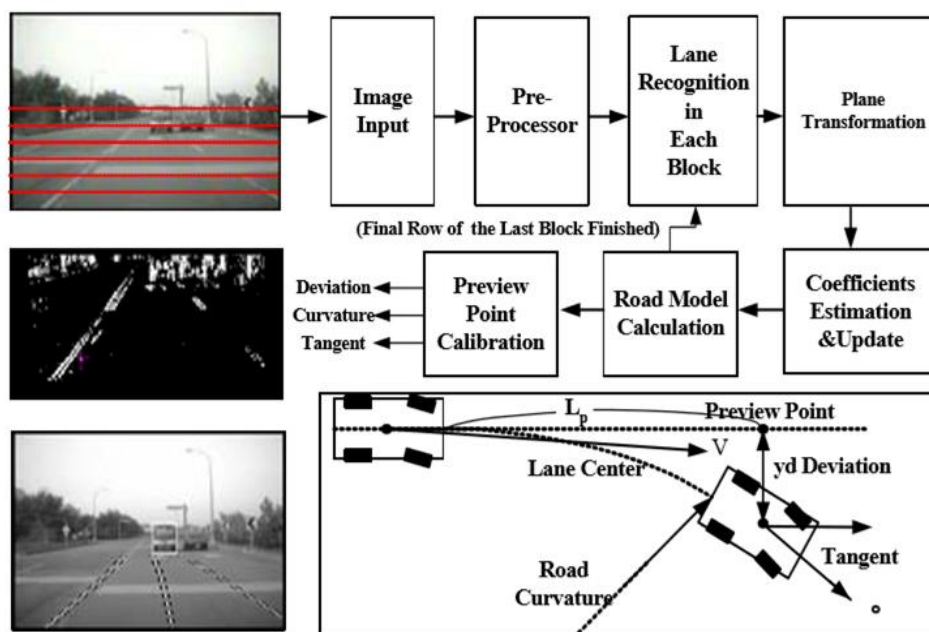


Fig. 2. Esquema algoritmo adaptativo [29]

Usando el punto de fuga para mejorar la detección de las marcas del carril de la carretera, se trabaja el desarrollo de la geometría, obteniendo un modelo plano de la carretera en el cual el carril queda definido por su centroide y orientación [28]. El algoritmo es implementado en el lenguaje de programación C++ con librerías de OpenCV. En este caso también modifican el punto de vista de la cámara con un cambio de coordenadas.

Profundizando en el cambio de coordenadas y el punto de vista de la cámara, se encuentra la perspectiva inversa de mapeado (IPM) y la IPM adaptativa [30], es decir, una perspectiva aérea de la imagen de la cámara que relaciona matemáticamente el sistema de coordenadas del mundo real y las perspectivas de la cámara. De esta manera, se pretende eliminar el efecto de perspectiva que se recibe de la cámara, facilitando el análisis y el cálculo de distancias reales [31]. La IPM adaptativa tiene también en cuenta la información del movimiento del vehículo para mejorar sus resultados.

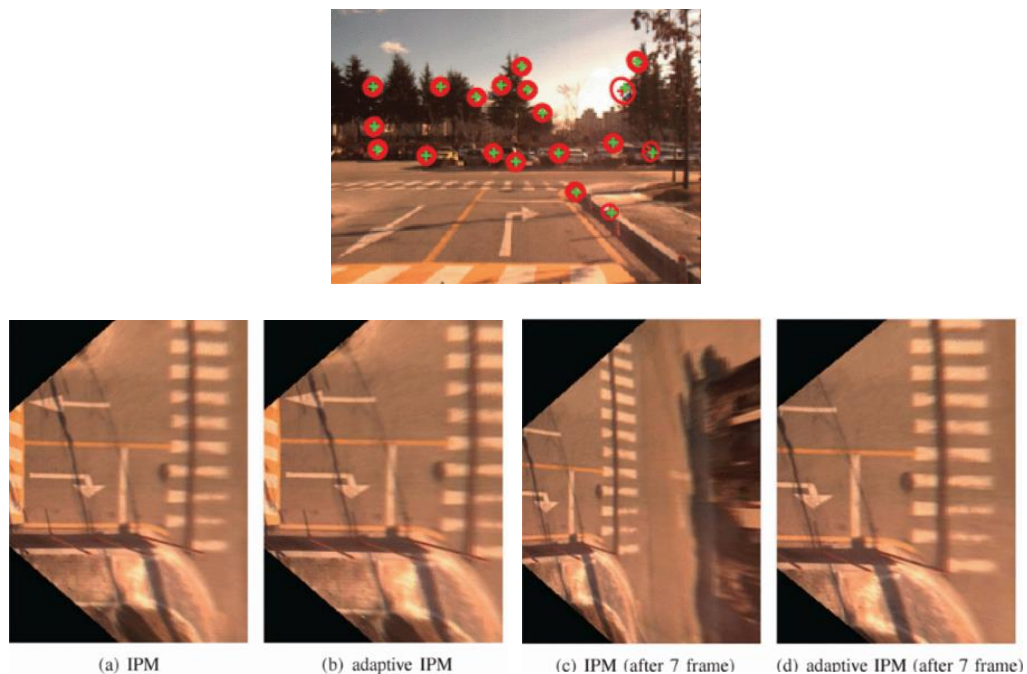


Fig. 3. Ejemplo IPM e IPM adaptativa [30]

Una alternativa para obtener la distancia del vehículo con respecto a las líneas del carril es un modelo para el movimiento lateral en términos de la curvatura del carril siguiendo los errores del estado del vehículo. Para una correcta localización se debe comparar las imágenes obtenidas de la cámara con la estimación realizada a partir del control lateral y obtener el error [32].

Para este tipo de modelos que estiman y corrigen los valores obtenidos, también se utiliza un filtro de Kalman extendido, UKF (unscented Kalman filter) [28] [33]. Se representa un modelo dinámico que obedece a la evolución de un sistema no lineal, estimando el vector de estado a priori aproximando las ecuaciones no lineales de estado y observación mediante la expansión de la serie de Taylor [34]. Siguiendo el polinomio de Taylor, se

puede expresar una función en términos de potencias x-a [35] para obtener un valor aproximado de la función alrededor de un punto [36].

Otro aspecto a tener en cuenta, como se ha podido observar en los estudios mencionados anteriormente, es la geometría de la carretera. Se parte de un análisis dinámico y cinemático de la carretera y del vehículo que permite obtener sus diferentes variables y se contempla una espiral de Euler o Clotoide [37] [32] para hallar la curvatura de la carretera, usando una aproximación polinómica a una función de tercer grado.

Como se ha visto, la conducción autónoma empieza a tener un largo recorrido, por lo que se pueden encontrar artículos menos recientes que utilizan un análisis más sencillo [38]. Usando dos cámaras, una para cada línea, y la velocidad del vehículo, se obtiene la desviación lateral de forma analítica.

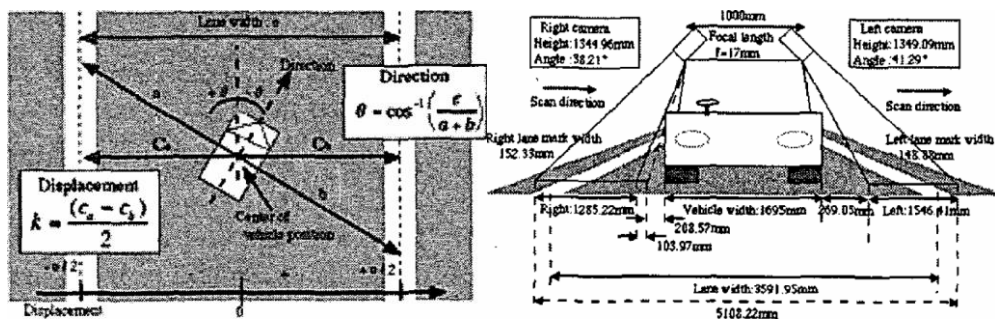


Fig. 4. Esquema cálculo analítico de la desviación angular y lateral [38]

También existen otros estudios más complejos que analizan el entorno y el resto de los carriles de la carretera. Estos suelen contar con un mayor número de recursos, así como una base de datos más amplia. Con una red Bayesiana, que permite crear un modelo gráfico probabilístico, se mejora la detección de los diferentes carriles de la carretera incluso cuando las líneas de estos no son tan visibles y localiza el carril en el que se encuentra el vehículo [39]. Además, se pueden usar los datos geométricos obtenidos previamente del mapa, los objetos detectados alrededor y la odometría del vehículo para posicionarlo correctamente en el mapa [40] o las herramientas que facilita OpenStreetMap [41].

2. ANÁLISIS Y DISEÑO DEL SISTEMA

2.1. Recursos iniciales

El Laboratorio de Sistemas Inteligentes de la Universidad Carlos III de Madrid cuenta con varias plataformas de investigación equipadas para probar los diferentes sistemas ADAS.

Una de estas plataformas, para la cual se desarrolla este Trabajo de Fin de Grado, es ADA (Autonomous Driving Automobile).



Fig. 5. Imagen de ADA

Este vehículo cuenta con los siguientes sensores para llevar a cabo la localización del automóvil en el carril y la percepción de su entorno:

- Cámara Basler instalada en la parte frontal del vehículo, con un sensor CMOS de 3.2MP y una óptica de 3.5 mm, con un campo de visión (FOV) frontal de 90°.
- LiDAR Robosense RS-LiDAR-32 de 32 planos, 360°.
- 2 LiDAR Robosense RS-LiDAR-16 de 16 planos, 360°.

En cuanto a la estructura de software, ROS Kinetic en Ubuntu 16.04 es la arquitectura de software elegida y operativa en el automóvil, por lo que será la usada en este proyecto para evitar posteriores errores de compatibilidad.

Para este trabajo se usan únicamente las imágenes obtenidas por el sistema estéreo de visión. En un principio se pensó utilizar también el LiDAR, de esta manera, combinando ambos sensores se podría saber la posición de cada píxel en las coordenadas reales de la carretera.

Finalmente, no se pudo contar con estos recursos debido a la complejidad de la calibración de la cámara con el LiDAR y la gran inversión de tiempo y personal que ello conlleva.

Por ello, se lleva a cabo la detección del carril únicamente con la cámara frontal del vehículo tras procesar las imágenes obtenidas para que muestren el carril en el que se encuentra el coche en color azul.

2.2. Entorno de simulación

Al igual que en el proyecto conjunto que se lleva a cabo en el Laboratorio de Sistemas Inteligentes de la Universidad Carlos III de Madrid, con relación al software instalado en el coche autónomo ADA, en este Trabajo de Fin de Grado se usa ROS Kinetic en Ubuntu 16.04 para evitar posteriores errores de compatibilidad.

2.2.1. ROS

ROS (Robot Operating System) [42] es un sistema operativo implementado para crear y desarrollar software para robots. Para ello provee librerías, herramientas y convenciones específicas que pretenden simplificar el complejo y robusto comportamiento de un robot a través de una variedad de plataformas.

Al ser un software colaborativo de código abierto, permite y favorece el trabajo conjunto de diferentes instituciones, laboratorios o proyectos individuales. ROS está diseñado de una manera modular, de manera que cada usuario pueda usar solo una parte específica de ROS que le interese e implementar sus propios módulos.

La estructura sobre la que está basada ROS es en el intercambio de mensajes entre Nodos. Un proyecto está formado por un número determinado de Nodos que publican mensajes de forma anónima en un asunto, también conocido como “topic” en inglés, al que otro nodo puede suscribirse y recibir estos mensajes.

Los mensajes están pensados para ser unidireccionales, pero también existen otros procedimientos en los que se recibe una respuesta a un requerimiento, es decir, un nodo provee un servicio al que otro nodo llamado cliente solicita y obtiene respuesta de vuelta.

Aunque se está trabajando para adaptar las librerías de ROS a otros sistemas operativos como Microsoft Windows, Debian, Arch, Mac Os X, etc. ROS está orientado para un sistema Ubuntu (Linux) con el que se trabaja en este proyecto.

Otra ventaja de ROS es poder utilizar diferentes lenguajes y librerías de programación, utilizando indistintamente C++ y Python, el que mejor se adapte al proyecto en cada momento. Por ejemplo, en este proyecto se utiliza en primer instancia Python con las librerías de OpenCV y posteriormente el proyecto empieza a desarrollarse en C++.

2.2.2. OpenCV

OpenCV (Open Source Computer Vision Library) [43] es una librería de software de código abierto para la visión por ordenador y el aprendizaje de máquinas. Provee una infraestructura común para estos propósitos y al igual que ROS se encuentra bajo la licencia de producto BSD, lo que permite modificarla y ser utilizada para fines comerciales.

Esta librería posee más de 2500 algoritmos optimizados entre los cuales encontramos algunos que permiten detectar y reconocer caras, identificar objetos, clasificar acciones humanas, seguir los movimientos de la cámara y los objetos que aparecen en ella, reconocer escenarios, etc. La red de usuarios de la comunidad de OpenCV está estimada en 47000 personas y más de 18 millones de descargas. Es usada comúnmente en compañías, grupos de investigación y cuerpos gubernamentales.

Existe OpenCV para C++, Python, Java y Matlab y soporta diferentes sistemas operativos como Windows, Linux, Android y Mac OS.

En este proyecto se utilizarán las librerías y funciones existentes en Python ya que es el lenguaje que predomina y en el que las librerías están más desarrolladas y son más sencillas.

2.3. Implementación

La implementación del programa contará con un paquete de ROS que lea y convierta las imágenes del video en un mensaje de ROS, y una serie de nodos organizados en cascada que procesen estos datos y obtengan las tres variables finales que se buscan en este proyecto: desviación angular, desplazamiento lateral izquierdo y desplazamiento lateral derecho del vehículo.

Al ser un sistema integrado en otro mayor de ayuda a la conducción en tiempo real, se utiliza una organización en cascada para llevar a cabo el proceso en un espacio de tiempo menor. Cada nodo implementa una parte del sistema y envía sus resultados al siguiente nodo, lo que permite trabajar con diferentes ramas en paralelo y no saturar un mismo nodo que realice todos los cálculos. De esta manera también se mejora la organización y se visualizan mejor los errores o modificaciones realizadas en cada nodo.

El código desarrollado para este proyecto se puede consultar en [GitHub](#).

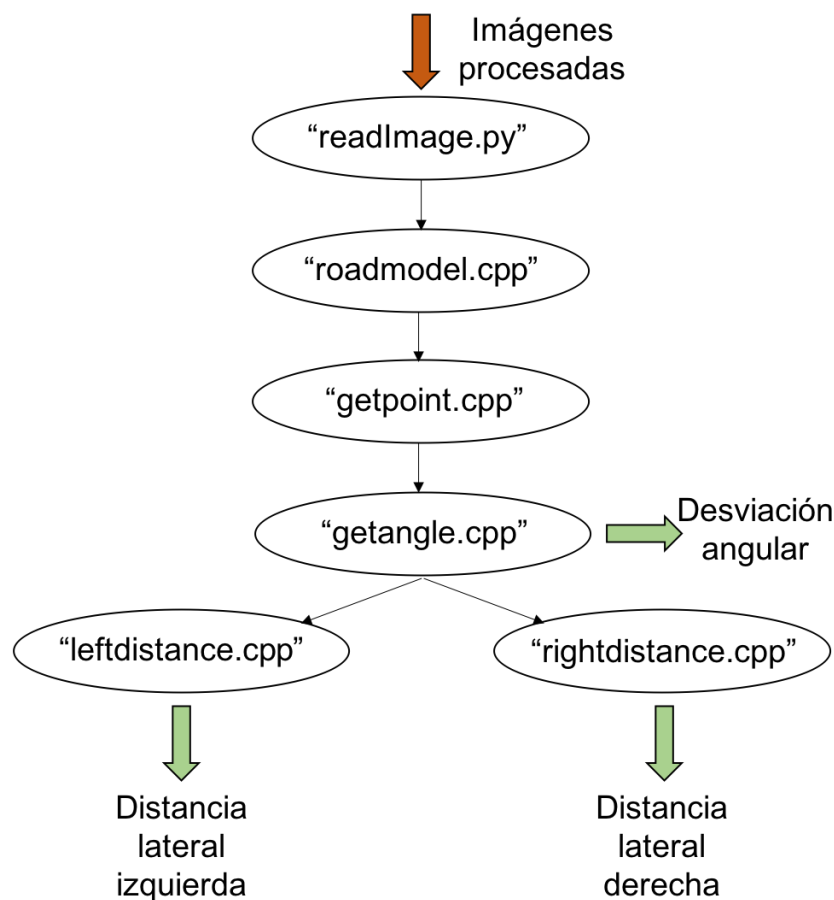


Fig. 6. Esquema organización del sistema

2.3.1. Obtención de los datos

En el video, que se recibe de la cámara instalada en el automóvil, es detectado previamente el carril en el que se encuentra el vehículo [44]. En este proyecto, las imágenes se procesan en primera instancia con un paquete de ROS que convierte y crea un mensaje para cada imagen del video.

Para la obtención de los datos necesarios de la imagen para desarrollar el sistema de obtención de las variables finales, se ha creado un nodo de ROS implementado con Python y librerías de OpenCV. El nombre de este nodo es “readImage.py”. Este nodo recibe las imágenes como mensaje de ROS.

Las imágenes sobre las que se trabaja marcan el carril en el que se encuentra el vehículo en color azul. Por ello, para llevar a cabo un filtro que delimite y permita obtener los puntos del carril de la carretera, se fijan los límites de color, usando el código RGB, que se corresponde con los azules.

Se evitará usar todo el espectro azul para tener el menor número de falsos positivos en la imagen que puedan llevar a error en un posterior análisis (cielo, personas, coches, señales de tráfico...).

En la parte inferior de la imagen se encuentra la coordenada donde se sitúa el ratón y el color RGB. Por ejemplo, los puntos de la siguiente imagen (Fig. 7.) tienen asociados el color correspondiente:

- 1: (x=241, y=443) ~ R:122 G:133 B:243
- 2: (x=244, y=376) ~ R:98 G:98 B:222
- 3: (x=464, y=382) ~ R:21 G:15 B:131
- 4: (x=193, y=324) ~ R:176 G:84 B:118
- 5: (x=529, y=111) ~ R:30 G:40 B:32
- 6: (x=155, y=53) ~ R:117 G:193 B:252



Fig. 7. Imagen de la carretera numerada

El rango de colores que se ha fijado para detectar los azules es:

- Límite inferior: (100, 120, 0)
- Límite superior: (140, 255, 250)

Posteriormente, se analiza cada imagen del video y se convierten estos valores RGB a un modelo de color HSV, lo que permite usar la función *cv2.inRange* y crear una máscara para el color azul que se ha delimitado.

$$mask = cv2.inRange(hsv, blueLowerLimit, blueUpperLimit)$$

Toda la imagen saldrá en color negro, a excepción del rango de azules definido, que se podrá ver de color blanco.



Fig. 8. Ejemplo conversión máscara blanco y negro

Otro recurso utilizado en este nodo, con motivo de mejorar la detección del carril, es la dilatación y erosión del color blanco de la máscara para evitar espacios vacíos no detectados dentro del carril y eliminar puntos azules ocasionales fuera de él (estos pueden ser debidos a reflejos, señales, personas...).

Al dilatar, el color blanco se expande rellenando los espacios vacíos no detectados, y al erosionar, los límites del carril vuelven a su estado original. Para eliminar los puntos azules pequeños detectados fuera del carril, se lleva a cabo el proceso contrario, primero se erosiona y a continuación se dilata.

En este nodo, se lleva a cabo en primera instancia una dilatación de 6 iteraciones mediante la función *cv2.dilate*. Este número de iteraciones permite la eliminación de huecos en el carril sin dilatar en un grado muy elevado y hacer que otros puntos azules externos al carril se unan a él.

mask = cv2.dilate (mask, None, iterations=6)



Fig. 9. Ejemplo dilatación de la máscara

A continuación, se lleva a cabo una erosión de 30 iteraciones de la máscara mediante la función `cv2.erode`. Ya que el carril supone la mayor superficie de azul y es la última en desaparecer por erosión, se lleva a cabo un gran número de iteraciones, haciendo desaparecer el resto de los elementos azules. Se evita un grado de erosión mayor para no permitir que los espacios vacíos que puedan quedar dentro del carril distorsionen el contorno de este.

Esto permite ampliar el rango de azules empleado, ya que, aunque detecten más objetos, mientras estos sean de menores dimensiones que el carril, desaparecerán posteriormente.

```
mask = cv2.erode (mask, None, iterations=30)
```



Fig. 10. Ejemplo erosión de la máscara

Finalmente, se lleva a cabo un último proceso de dilatación de 24 iteraciones. El total de iteraciones de erosión y dilatación deben ser iguales para respetar el orden de magnitud inicial y no alterar las dimensiones del carril en la imagen.

$$mask = cv2.dilate (mask, None, iterations=24)$$

Para evaluar gráficamente el carril detectado que se usa para el posterior análisis y cálculo de las variables de interés, se dibuja el contorno de este en la imagen (Fig.12.).



Fig. 11. Ejemplo máscara final



Fig. 12. Ejemplo detección contorno del carril

A continuación, se da paso a la extracción de los puntos que delimitan el carril, tanto por la izquierda como por la derecha.

En este proceso, se guarda para cada coordenada y , dos coordenadas x , izquierda (x_i) y derecha (x_d). Es decir, para cada coordenada y , se obtienen las coordenadas x que delimitan el carril por ambos lados.

Las coordenadas de la imagen comienzan en la esquina superior izquierda, donde se sitúa la coordenada (0, 0). La coordenada y aumenta a medida que se baja en la imagen, y la coordenada x aumenta de izquierda a derecha.

Estos datos se pueden obtener de forma sencilla utilizando un bucle. Para cada fila (coordenada y) se analiza el color de cada píxel hasta encontrar uno blanco (igual a 255 en la máscara de la imagen en blanco y negro), se guardan las coordenadas de este píxel, y se pasa a analizar la fila siguiente. Como para cada fila se necesitan dos valores, se lleva a cabo el mismo proceso empezando primero el análisis por el primer valor de x (izquierda) y luego por el último (derecha).

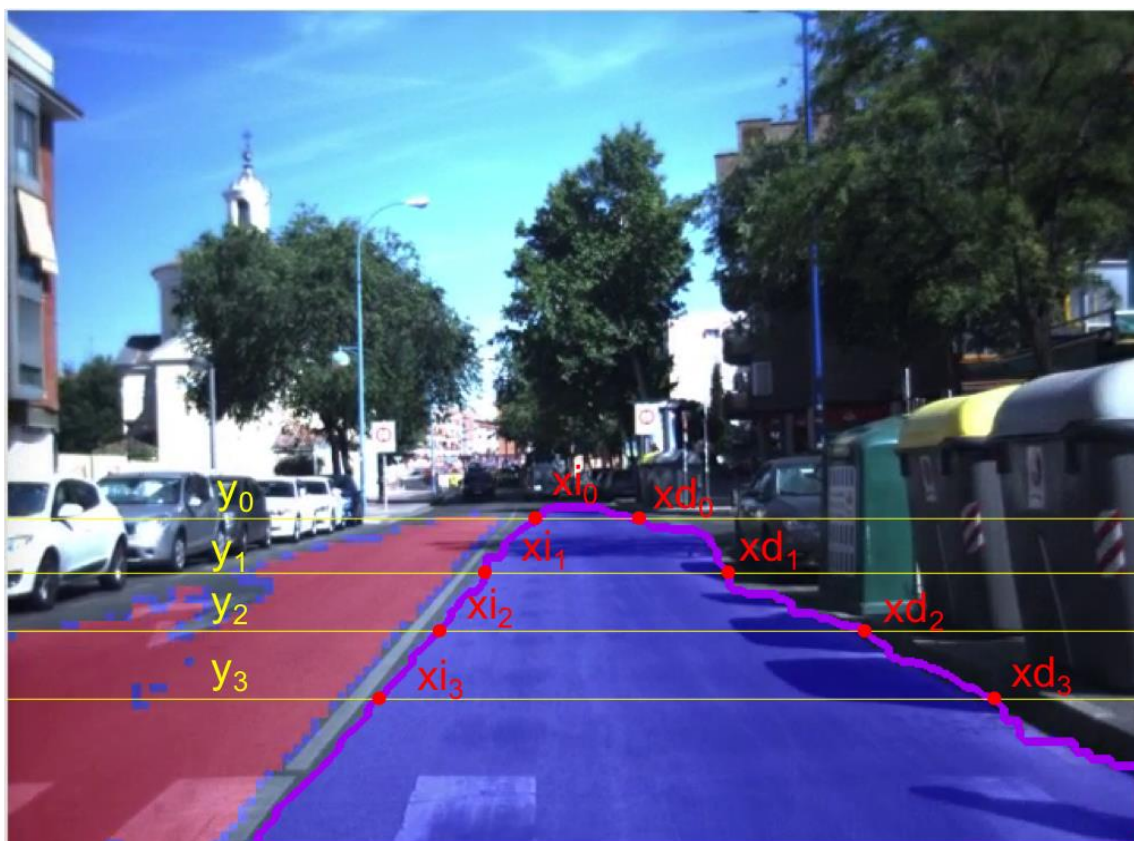


Fig. 13. Ejemplo obtención datos del carril

Para evitar un exceso de datos, agilizar el proceso y eliminar posibles azules detectados en el marco superior de la imagen (cielo principalmente, ya que es la segunda mayor

superficie azul y no siempre desaparece por completo, en ocasiones, es la superficie de mayor tamaño), se comienza el análisis en

$$y = h \cdot 0,4 \quad (1)$$

es decir, la dimensión de altura (h) de la imagen multiplicada por $0,4$. De esta manera, se comienza a analizar la imagen por un valor de y ligeramente inferior (más arriba en la imagen) al que delimita el horizonte de la carretera en condiciones normales, suponiendo que la pendiente de esta sea igual a la del vehículo, y se va aumentando hasta llegar al mayor valor de y (límite inferior de la imagen).

Estos datos se guardan en una matriz y se preparan de cara a obtener el modelo de la carretera. Es decir, se filtra y elige los datos necesarios que publica en un mensaje de ROS para que el siguiente nodo desarrolle el modelo de la carretera.

Para desarrollar los diferentes modelos de carretera y sus curvaturas se necesitan tres o cuatro puntos dependiendo del grado del polinomio y el número de parámetros. Por ello, se hace una distinción de este nodo para cada modelo y se crean “*readImage_clo.py*” y “*readImage_par.py*”.

Estos tres o cuatro puntos se obtienen eligiendo tres o cuatro coordenadas y y sus dos respectivas coordenadas x .

Para el modelo de la carretera desarrollado para la desviación angular se utilizan los puntos que pasan por el centro del carril.

Para cada y ,

$$x_c = \frac{x_i + x_d}{2} \quad (2)$$

Estas coordenadas y se eligen de manera que cubran de la mejor manera posible la curvatura del carril y estén lo más distanciadas entre sí. Por ello, estos puntos tienden a ser más próximos al horizonte, donde cambia más rápidamente la curvatura del carril debido a la perspectiva de la imagen. También se evitan los primeros y últimos valores de y . Los primeros debido a que en el horizonte de la carretera x_i y x_d tienden a juntarse, siendo más difusas las líneas que delimitan los carriles y generan un mayor error. Los últimos debido a que la desviación angular o lateral del vehículo puede hacer que las líneas que delimiten el carril corten la imagen por uno de los laterales y estos distorsionen el modelo de la carretera.

Los puntos elegidos, llamando d a la distancia entre el horizonte de la carretera (primer píxel del carril detectado) y el límite inferior de la imagen, son, dependiendo del modelo de carretera:

Para un modelo cuadrático o parabólico,

$$y_0 = 0,10 d \quad (3)$$

$$y_1 = 0,30 d \quad (4)$$

$$y_2 = 0,75 d \quad (5)$$

Para un modelo cúbico o clotoïdial,

$$y_0 = 3 \quad (6)$$

$$y_1 = 0,15 d \quad (7)$$

$$y_2 = 0,40 d \quad (8)$$

$$y_3 = 0,80 d \quad (9)$$

Estas variables son definidas como *int* (integer), por lo que siempre serán un número entero para poder relacionarlos directamente con sus píxeles correspondientes.

Una vez se tienen las coordenadas y elegidas para cada caso y sus respectivas x_i , x_d y x_c , son publicadas bajo el asunto “*RoadPoints*”. Estas coordenadas serán utilizadas por los nodos que obtienen los parámetros de la curva para los diferentes modelos de carretera.

2.3.2 Desarrollo modelo de carretera

En este proyecto se analizan diferentes modelos de curvatura propuestos por distintos estudios para ver cual se adapta mejor y reduce el porcentaje de error. Se analizará un Clotoïde o Espiral de Euler [32] [37] y un modelo cuadrático o parabólico [29].

Se han implementado dos nodos diferentes, cada uno para un modelo de carretera, para calcular los parámetros de la curva. Las coordenadas (x_n, y_n) se obtienen al suscribirse al

asunto “*RoadPoints*” del nodo anterior “*readImage.py*”. Estos nodos están implementados en C++ y, a partir de aquí, todos los nodos son desarrollados en C++, ya que ofrece un mejor rendimiento que Python, lo que permite que el proceso se pueda ejecutar con mayores garantías en tiempo real.

2.3.2.1 Modelo cuadrático o parabólico

Este modelo propone una parábola para definir la trayectoria de la carretera.

Una parábola es la curva resultante de cortar un cono recto con un plano cuyo ángulo de inclinación respecto al eje de revolución del cono sea igual al presentado por su generatriz, siendo paralelo a esta recta [45].

Se puede definir una parábola mediante una ecuación de segundo grado.

El nodo “*roadmodel_par.cpp*”, es el nodo específico para calcular las constantes de una parábola. Al ser una ecuación de segundo grado, se necesitan tres puntos de la carretera y un sistema de tres ecuaciones para calcular las constantes.

$$y_0 = C_0 + C_1x_0 + C_2x_0^2 \quad (10)$$

$$y_1 = C_0 + C_1x_1 + C_2x_1^2 \quad (11)$$

$$y_2 = C_0 + C_1x_2 + C_2x_2^2 \quad (12)$$

2.3.2.2 Modelo cúbico o Clotoidal

El radio de curvatura de un clotoide disminuye de forma inversamente proporcional a la distancia que se recorre sobre ella. Por ello, en el diseño y construcción de carreteras y vías de ferrocarril, se utiliza como curva de transición entre una línea recta y una curva. Esto proporciona una mayor distancia de visibilidad y permite al vehículo un cambio de trayectoria más suave [46].

En este proyecto se usará una aproximación a una función cúbica como clotoide al igual que en el estudio realizado por Seung-Hi Lee y Chung Choo Chung [32].

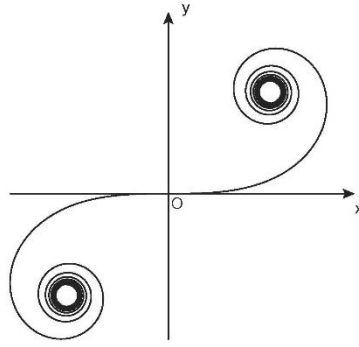


Fig. 14. Ejemplo de clotoide [47]

El nodo “*roadmodel_clo.cpp*”, es el nodo específico para calcular las variables de un clotoide. Al ser una ecuación de tercer grado, se necesitan cuatro puntos de la carretera y un sistema de cuatro ecuaciones para calcular las constantes.

$$y_0 = C_0 + C_1x_0 + C_2x_0^2 + C_3x_0^3 \quad (13)$$

$$y_1 = C_0 + C_1x_1 + C_2x_1^2 + C_3x_1^3 \quad (14)$$

$$y_2 = C_0 + C_1x_2 + C_2x_2^2 + C_3x_2^3 \quad (15)$$

$$y_3 = C_0 + C_1x_3 + C_2x_3^2 + C_3x_3^3 \quad (16)$$

2.3.2.3 Obtención de parámetros de la carretera

Para resolver estos sistemas de ecuaciones y obtener las constantes de la curva, se crea una matriz a partir de las coordenadas recibidas. Este sistema de ecuaciones expresado en forma de matriz es:

Para un modelo parabólico,

$$\begin{pmatrix} 1 & x_0 & x_0^2 & y_0 \\ 1 & x_1 & x_1^2 & y_1 \\ 1 & x_2 & x_2^2 & y_2 \end{pmatrix} \quad (17)$$

```

The matrix we are solving:
1      316.5      100172      286
1      357.5      127806      376
1      327.5      107256      322

```

Fig. 15. Ejemplo matriz sistema parabólico

Para un modelo clotoidal,

$$\begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 & y_0 \\ 1 & x_1 & x_1^2 & x_1^3 & y_1 \\ 1 & x_2 & x_2^2 & x_2^3 & y_2 \\ 1 & x_3 & x_3^2 & x_3^3 & y_3 \end{pmatrix} \quad (18)$$

```

The matrix we are solving:
1      305      93025      2.83726e+07      267
1      350.5      122850      4.3059e+07      354
1      331.5      109892      3.64293e+07      324
1      318      101124      3.21574e+07      294

```

Fig. 16. Ejemplo matriz modelo clotoidal

El siguiente paso será resolver la matriz correspondiente mediante el método de Gauss-Jordan.

Para un modelo parabólico,

```

The matrix after gauss-elimination is:
1      316.5      100172      286
0      41      27634      90
0      0      -330      11.8537

```

Fig. 17. Ejemplo matriz resuelta modelo parabólico

Para un modelo clotoïdial,

```
The matrix after gauss-elimination is:  
1      305      93025      2.83726e+07      267  
0      45.5      29825.2      1.46864e+07      87  
0     -3.55271e-15      -503.5      -496954      6.32967  
0     2.98117e-15      0      5703.75      -3.16853
```

Fig. 18. Ejemplo matriz resuelta modelo clotoïdial

De esta manera se obtienen las constantes de la ecuación que se toman como modelo de carretera. Estos parámetros son publicados como mensaje de ROS bajo el asunto “*parameters*”.

Para un modelo parabólico,

```
The values of the variables are:  
C0 = -4473.08  
C1 = 26.4053  
C2 = -0.0359202  
  
Road Model Equation: Y = -4473.08 + 26.4053*X + -0.0359202*X2
```

Fig. 19. Ejemplo parámetros de la carretera, modelo parabólico

Para un modelo clotoïdial,

```
The values of the variables are:  
C0 = 18026.4  
C1 = -169.947  
C2 = 0.535725  
C3 = -0.000555518  
  
Road Model Equation: Y = 18026.4 + -169.947*X + 0.535725*X2 + -0.000555518*X3
```

Fig. 20. Ejemplo parámetros de la carretera, modelo clotoïdial

2.3.3 Obtención de un punto de la curva

Detrás de cada modelo de carretera (“*roadmodel.cpp*”), se implementa un nodo que se suscribe a los parámetros de la carretera obtenidos en este, es decir, al asunto “*parameters*” y procede a calcular la coordenada x en una determinada y . Esta coordenada

y es la correspondiente a aquella que se encuentre unos pocos metros por delante del automóvil, pero podría variar en cuanto a la necesidad de la velocidad del vehículo. A mayor velocidad, a mayor distancia se encontrará este punto del vehículo.

Estos nodos son específicos para cada modelo de carretera, dependiendo de si la ecuación de la curva es un polinomio de segundo o tercer grado, “*getpoint_par.cpp*” o “*getpoint_clot.cpp*” respectivamente. Los resultados obtenidos son publicados bajo el asunto “*PredictedPoint*”.

2.3.3.1 Modelo cuadrático o parabólico

En el caso de un modelo cuadrático, la ecuación a resolver es

$$0 = C_0 + C_1x + C_2x^2 \quad (19)$$

Para resolver estas ecuaciones de segundo grado se utiliza la ecuación general de ecuaciones de segundo grado.

Si escribimos la ecuación de segundo grado como

$$ax^2 + bx + c = 0 \quad (20)$$

combinada con los coeficientes que se obtienen del modelo de la carretera

$$a = C_2 \quad (21)$$

$$b = C_1 \quad (22)$$

$$c = C_0 - y_1 \quad (23)$$

Se pueden obtener los diferentes resultados de x como

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (24)$$

De esta manera, se obtienen dos raíces como solución. Gráficamente se puede comprobar como una de ellas se adapta mejor a la solución real, siendo la otra raíz totalmente incoherente.

Para analizar gráficamente las raíces obtenidas se obtienen todas ellas para el rango de datos y que se adquieren en primera estancia con el primer nodo *readImage.py*. Se comparan los resultados conseguidos con la ecuación del modelo de carretera y los datos sacados directamente de la imagen.



Fig. 21. Ejemplo imagen de la carretera

Por ejemplo, para la imagen anterior (Fig. 21.), se obtienen los siguientes resultados (Fig. 22.), siendo x_c las coordenadas reales extraídas de la imagen, y XC1 y XC2, las diferentes soluciones obtenidas a partir de la ecuación del modelo de carretera:

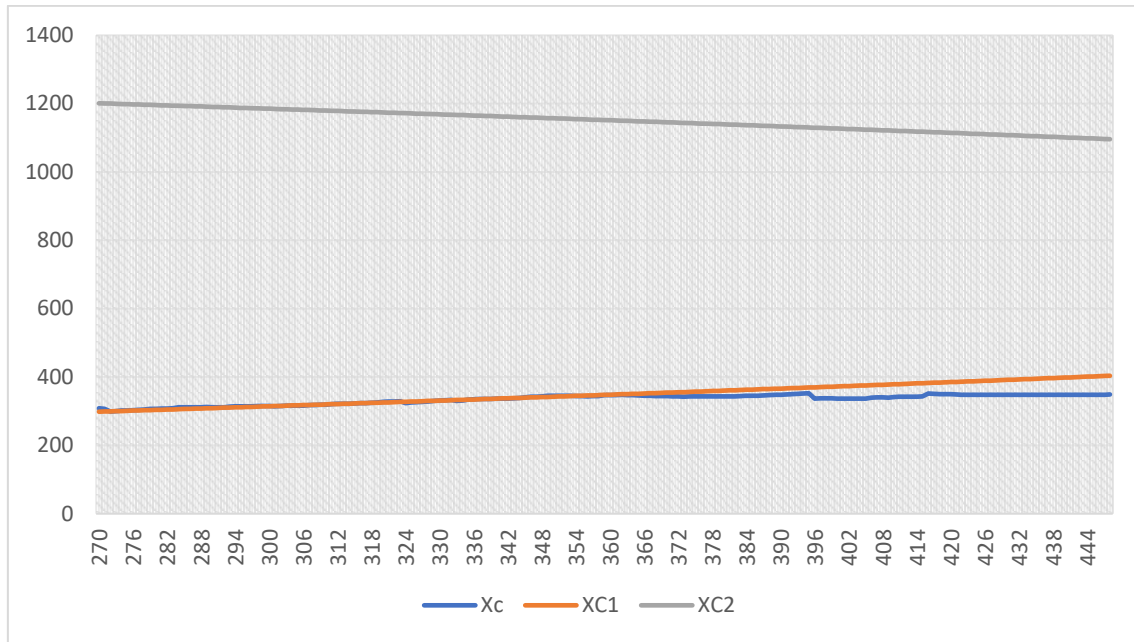


Fig. 22. Ejemplo resultados ecuación modelo parabólico

Se puede comprobar que los resultados que mejor se adaptan vienen dados por la raíz XC1, por el contrario, la solución de la raíz XC2 se desvía totalmente de la trayectoria del carril. Sin embargo, estos resultados pueden variar y no siempre se adapta mejor XC1 que XC2. Más adelante, en el apartado 2.3.3.3. Soluciones al modelo de carretera, se detalla el modo en el que el sistema selecciona el resultado más acorde en cada circunstancia.

2.3.3.2 Modelo cúbico o clotoidal

En el caso del clotoide la ecuación a resolver es:

$$0 = C_0 + C_1x + C_2x^2 + C_3x^3 \quad (25)$$

Para resolver estas ecuaciones de tercer grado se utiliza una aproximación discriminante [48].

Escribiendo la ecuación de tercer grado como

$$ax^3 + bx^2 + cx + d = 0 \quad (26)$$

Combinados con los coeficientes que se obtienen del modelo de la carretera

$$a = C_3 \quad (27)$$

$$b = C_2 \quad (28)$$

$$c = C_1 \quad (29)$$

$$d = C_0 - y \quad (30)$$

Se calculan los discriminantes de 0 y 1.

$$\Delta_0 = b^2 - 3ac \quad (31)$$

$$\Delta_1 = 2b^3 - 9abc + 27a^2d \quad (32)$$

y una vez se tienen estos valores, se calcula el discriminante cúbico

$$\Delta = \frac{\Delta_1^2 - 4\Delta_0^3}{-27a^2} \quad (33)$$

Si $\Delta > 0$, la ecuación tiene tres raíces reales distintas.

Si $\Delta = 0$, la ecuación tiene raíces múltiples y todas reales. Puede ser una raíz triple o una doble y una simple.

Si $\Delta < 0$, la ecuación tiene una raíz real y dos raíces complejas.

Antes de calcular las raíces, es necesario obtener las constantes c y u ,

$$c = \sqrt[3]{\frac{\sqrt{\Delta_1^2 - 4\Delta_0^3} + \Delta_1}{2}} \quad (34)$$

$$u = \frac{-1 + \sqrt{3}}{2} \quad (35)$$

Las soluciones a la ecuación cúbica vienen dadas por la siguiente formula,
para $n = 1, 2$ y 3 .

$$x_n = \frac{-\left(\frac{b+u^n c+\Delta_0}{u^n c}\right)}{3a} \quad (36)$$

Existen tres posibles soluciones para cada coordenada y .

Para analizar las raíces obtenidas gráficamente, al igual que en el modelo anterior, se obtienen todas ellas para el rango de datos y que se adquieren directamente de la imagen pertenecientes al carril y se comparan los resultados.

Para un clotoide como modelo de carretera en la imagen de la Fig. 21. se obtienen los siguientes resultados (Fig. 23.) siendo x_c las coordenadas reales extraídas de la imagen, y XC1, XC2 y XC3, las diferentes soluciones obtenidas a partir de la ecuación del modelo de carretera:

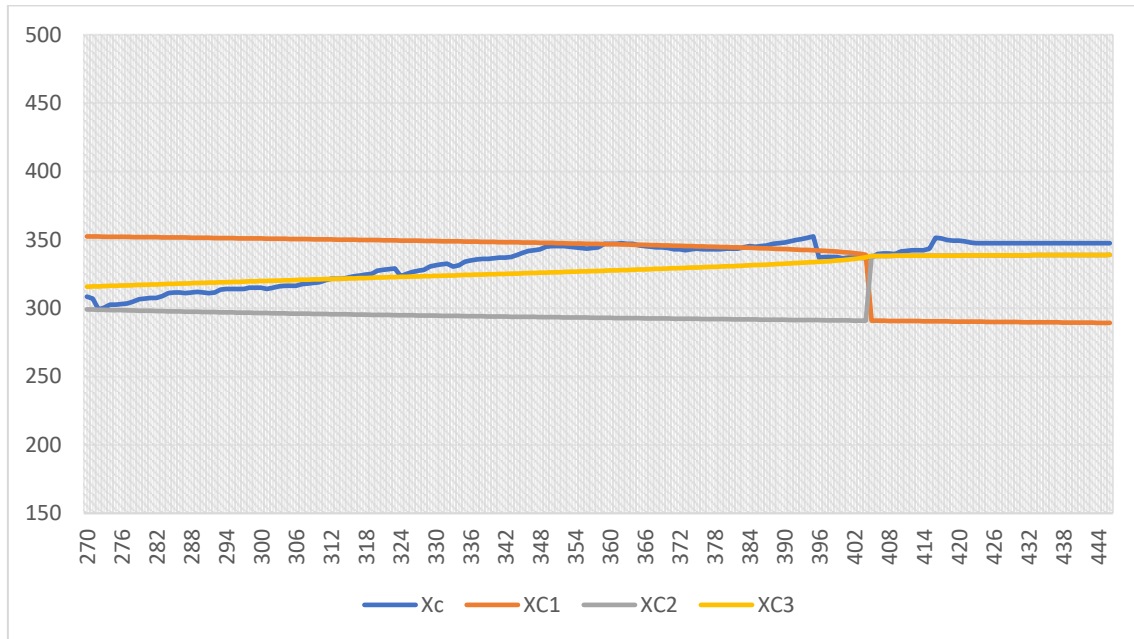


Fig. 23. Ejemplo resultados ecuación modelo clotoideal

Aunque, la diferencia entre las distintas soluciones no es tan evidente como en el modelo anterior, se puede apreciar en todo momento como una de ellas predomina sobre las demás. En este caso, se puede observar como XC3 se adaptaría mejor a la línea central del carril. A continuación, se detalla la manera en la que el sistema selecciona la más validez entre estas tres posibles soluciones.

2.3.3.3 Soluciones al modelo de carretera

Dependiendo de los parámetros del modelo de carretera, se tienen en cuenta las soluciones de las diferentes raíces y se escoge aquella que mejor se adapte a los datos reales para cada caso. Esta cambia para cada imagen.

Para determinar estos resultados en el sistema, se compara la suma total de las distancias (*dist*) entre las coordenadas utilizadas para desarrollar el modelo de carretera y los puntos x obtenidos mediante las diferentes soluciones del modelo para dichas coordenadas y . Las soluciones de la raíz con menor distancia entre las coordenadas obtenidas y las reales es la que se tiene en cuenta para el desarrollo posterior del sistema.

La variable k dependerá del modelo de carretera, ya que para un modelo parabólico se usan 3 puntos para determinar su ecuación ($k=2$) y en un modelo clotoïdial se utilizan 4 puntos ($k=3$).

$$dist = \sum_{n=0}^k |x_n - X_n| \quad (37)$$

2.3.4. Desviación angular

Una vez se predice el punto de la curva por el cual pasa el centro del carril, se procede a calcular la orientación del carril en dicho punto.

Para ello se implementa el nodo “*getangle.cpp*”, que parte de los datos obtenidos en el nodo anterior “*getpoint.cpp*”. Al igual que los nodos anteriores, se implementa un nodo diferente para cada modelo.

La recta tangente a la curva del modelo de carretera en el punto predicho permite conocer la dirección del carril en estas coordenadas. El ángulo que se forma entre la dirección del carril y la dirección del vehículo será la primera variable inicial propuesta para el control del automóvil: la desviación angular del vehículo respecto del carril.

La recta tangente a una curva en un punto concreto viene determinada por:

$$y - f(x_4) = f'(x_4)(x - x_4) \quad (38)$$

siendo $f(x_4)$ la imagen de la curva para la coordenada x predicha, es decir, y_4 .

La pendiente de la recta tangente es la derivada de la curva en dicho punto

$$m = f'(x_4) \quad (39)$$

para un modelo parabólico,

$$m = c_1 + 2 c_2 x_4 \quad (40)$$

para un modelo clotoïdial,

$$m = c_1 + 2 c_2 x_4 + 3 c_3 x_4^2 \quad (41)$$

Por lo tanto, la recta tangente es

$$y = mx + cn \quad (42)$$

donde

$$cn = -mx_4 + y_4 \quad (43)$$

Una vez obtenida la recta tangente, se calcula el punto de esta que corta con la dirección del vehículo en ese preciso instante.

Suponiendo que la dirección del vehículo sigue una trayectoria recta y siendo w el ancho total de la imagen, ésta se puede definir como:

$$y = \frac{w}{2} \quad (44)$$

Entonces, el punto en el que se cortan ambas rectas es (x_t, y_t) , donde

$$x_t = \frac{w}{2} \quad (45)$$

$$y_t = mx_t + cn \quad (46)$$

Formando un triángulo rectángulo con una recta perpendicular a la dirección del vehículo que pase por el punto (x_4, y_4) de la curva del carril, es decir,

$$f(x) = y_4 \quad (47)$$

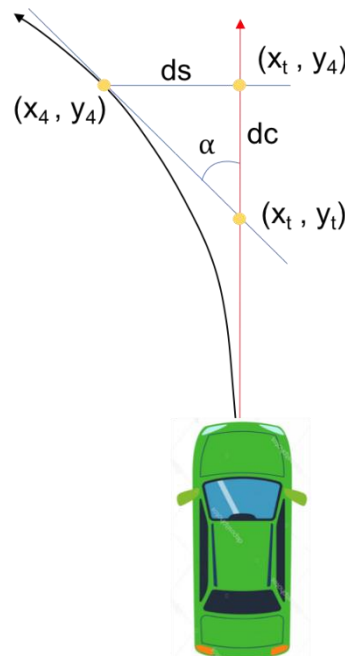


Fig. 24. Esquema calculo desviación angular

Se calculan las distancias entre estos puntos.

$$ds = |(x_4, y_4), (x_t, y_t)| = \sqrt{(x_t - x_4)^2} \quad (48)$$

$$dc = |(x_t, y_4), (x_t, y_t)| = \sqrt{(y_t - y_4)^2} \quad (49)$$

Finalmente, la desviación angular es

$$\alpha = \tan^{-1} \left(\frac{ds}{dc} \right) \quad (50)$$

Debido a la obtención de un número positivo en la ecuación anterior (50), se tiene en cuenta el signo de la pendiente de la recta tangente para determinar el sentido de la desviación angular.

Como resultado final, si la desviación angular es positiva, la trayectoria del vehículo se estaría desviando hacia la derecha con relación a la trayectoria seguida por la carretera, es decir, habría que realizar un giro a la izquierda para corregir la posición. Por el contrario, si la desviación angular es negativa, el vehículo se estaría desviando a la izquierda.

2.3.5. Desplazamiento lateral

Tras la obtención de la desviación angular, se procede al cálculo de las distancias laterales del automóvil respecto de las líneas del carril. Se halla paralelamente el desplazamiento lateral izquierdo y derecho en diferentes nodos para minimizar el tiempo de ejecución del algoritmo, “*leftdistance.cpp*” y “*rightdistance.cpp*”, y al igual que en los nodos anteriores, se desarrollan nodos diferentes para cada modelo de carretera.

Estas distancias son perpendiculares a la trayectoria del carril, por lo que se usa el ángulo (α) obtenido previamente suscribiéndose al asunto “*anglepredicted*”.

En estos nodos, al igual que en los apartados 2.3.2. Desarrollo del modelo de carretera y 2.3.3. Obtención de un punto de la curva, se lleva a cabo el desarrollo y obtención de los puntos de las líneas que delimitan el carril por ambos lados.

En este caso, aunque no sea necesaria la curva para hallar la recta tangente, se utiliza este proceso para minimizar el error en caso de existir ruido en la detección del carril y poder utilizar en un momento dado algún punto que la cámara no ha podido captar. Por ejemplo, en ocasiones donde las líneas que delimitan el carril no cortan la imagen por su límite inferior.

Para este proceso se obtienen los parámetros de la curva que forman las líneas que delimitan el carril utilizando las mismas x_i (para el cálculo de coordenadas de la línea izquierda) y x_d (para el cálculo de coordenadas de la línea derecha) que se usan en nodos anteriores para calcular x_c .

Ya que para evaluar los resultados del sistema se utiliza una imagen de video obtenida sobre un entorno urbano, en el que la velocidad no es elevada y está limitada, en este trabajo, se obtienen las distancias laterales para la misma coordenada y para la que se halla la desviación angular. Aunque, de igual modo se podrían obtener a una distancia superior o menor, cuanto mayor sea la distancia escogida, dependiendo de la curvatura de la carretera y de la desviación angular y lateral, mayor error se generará.

Las nuevas coordenadas obtenidas son (x_5, y_5) .

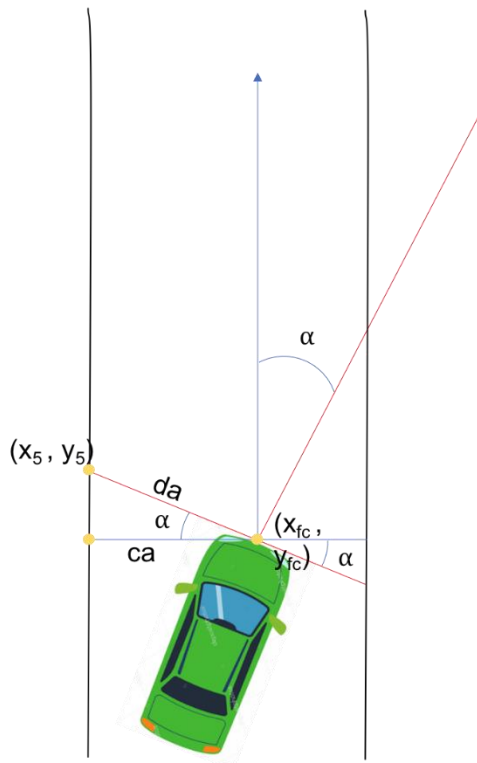


Fig. 25. Esquema cálculo desplazamiento lateral

Si se define el punto de referencia de la parte frontal central del vehículo como (x_{fc}, y_{fc}) , da es la distancia entre (x_{fc}, y_{fc}) y (x_5, y_{fc}) .

Por lo tanto

$$da = |(x_{fc}, y_{fc}), (x_5, y_5)| = \sqrt{(x_{fc} - x_5)^2} \quad (51)$$

Finalmente, la distancia lateral (ca), perpendicular a la trayectoria de la carretera, de la línea que delimita el carril al punto frontal central del automóvil (x_{fc}, y_{fc}) es

$$ca = da \cos \alpha \quad (52)$$

2.3.6. Visualización

Una vez calculadas las tres variables planteadas en este proyecto, se ha añadido un último nodo llamado “display.py” que recoge los resultados obtenidos y los muestra en la pantalla. Al trabajar en este nodo directamente con las imágenes de video obtenidas desde la cámara del vehículo, se vuelve a utilizar el lenguaje de programación Python con las librerías de OpenCV.

De esta manera, se pueden ver las imágenes generadas por la cámara con la detección del carril y un gráfico que indica la desviación angular y el desplazamiento lateral del vehículo, mostrando cuantitativamente la distancia que se desvía el automóvil hacia uno de los laterales respecto del centro del carril.

En el ejemplo a continuación, en la figura 25, fijándose en el gráfico que aparece en la pantalla, el vehículo se estaría desviando $9,5^\circ$ a la izquierda, por lo que se tendría que realizar un giro a la derecha para corregir su trayectoria. Por otro lado, el automóvil estaría desplazado un 42% hacia la izquierda con respecto al centro del carril. La barra azul indica la distancia en porcentaje que habría que recorrer al lado contrario para situarse en el centro del carril.

Este porcentaje (D), teniendo en cuenta las distancias laterales, izquierda (D_i) y derecha (D_d) halladas previamente, se calcula mediante

$$D(\%) = 2 \times \left(\frac{D_i}{D_i + D_d} - 0,5 \right) \times 100 \quad (53)$$

En caso de que D sea negativa, el vehículo estará a la izquierda del carril. Por el contrario, si es positiva, se encontrará a la derecha de este.

De esta forma, se pretende mostrar gráficamente el valor de las variables para que resulte más sencillo el análisis de los resultados y conocer la efectividad del control lateral del vehículo en todo momento.

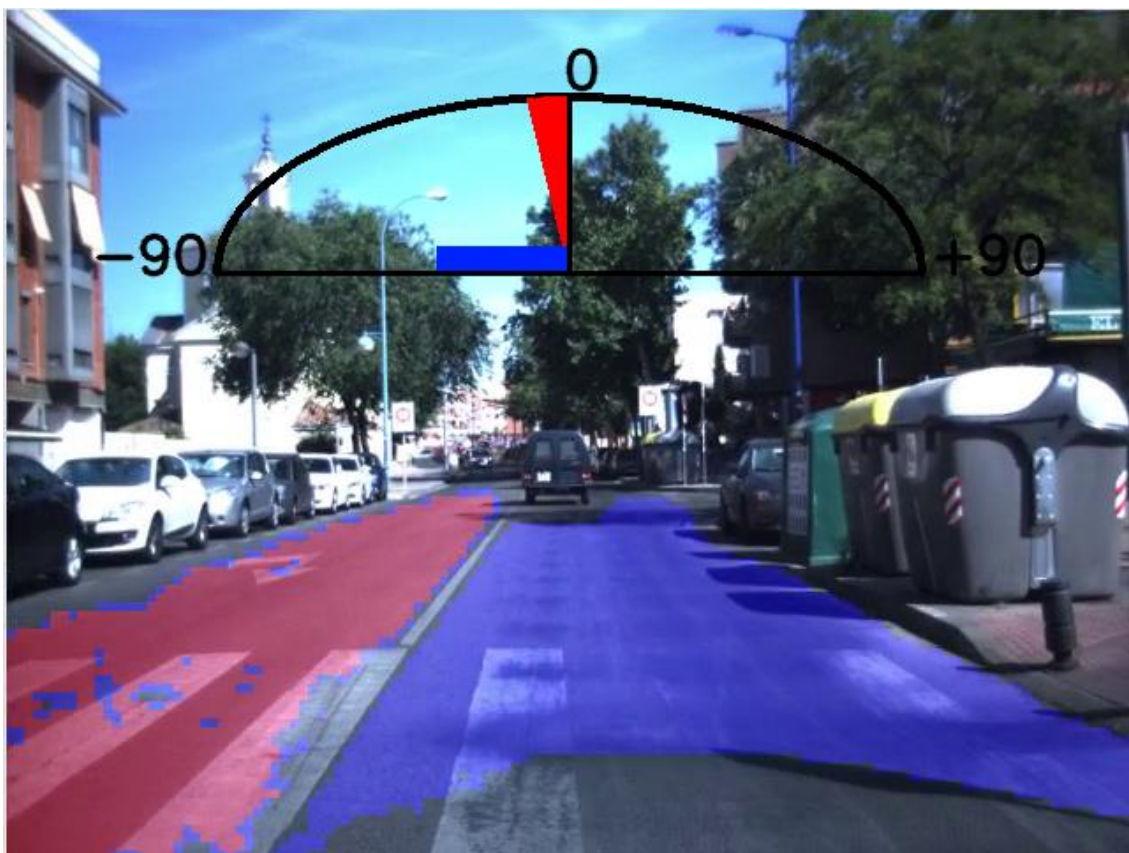


Fig. 26. Ejemplo de visualización de resultados en la pantalla

3. RESULTADOS

En este apartado se exponen los resultados del sistema implementado, comparando los modelos de carretera parabólico y clotoïdial.

Para analizar los resultados se usarán las imágenes obtenidas directamente de la cámara frontal del vehículo ADA, proporcionado por la Universidad Carlos III de Madrid, en un recorrido por las calles cercanas a su campus de Leganés.

Se analizarán de manera cualitativa, ya que no se dispone de una base de datos que permita conocer las distancias y trayectorias reales de la carretera. Para ello, se comparan gráficamente los resultados obtenidos con las ecuaciones de los diferentes modelos de carretera para todo el espectro y del carril sobre las imágenes. Los resultados son mostrados en verde. La línea roja es la trayectoria que seguiría el vehículo en caso de mantener la dirección actual.

Se obtienen los siguientes resultados:

Imagen 1



Fig. 27. Resultado imagen 1 modelo parabólico



Fig. 28. Resultado imagen 1 modelo clotoïdial

Imagen 2

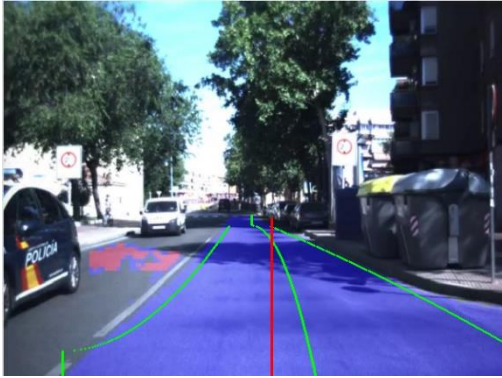


Fig. 29. Resultado imagen 2 modelo parabólico

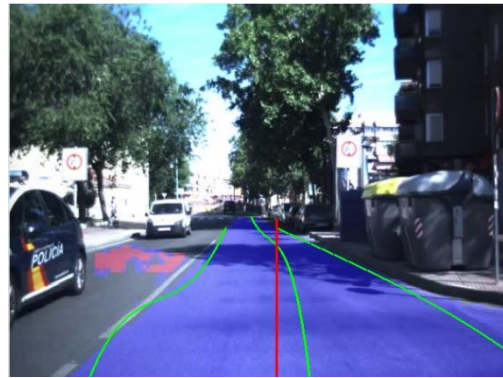


Fig. 30. Resultado imagen 2 modelo clotoidal

Imagen 3

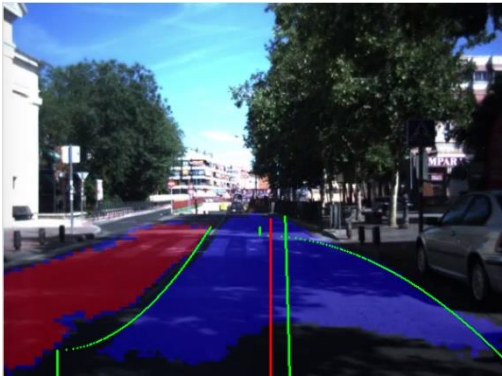


Fig. 31. Resultado imagen 3 modelo parabólico

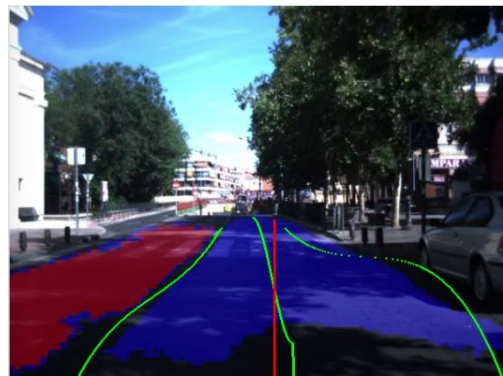


Fig. 32. Resultado imagen 3 modelo clotoidal

Imagen 4

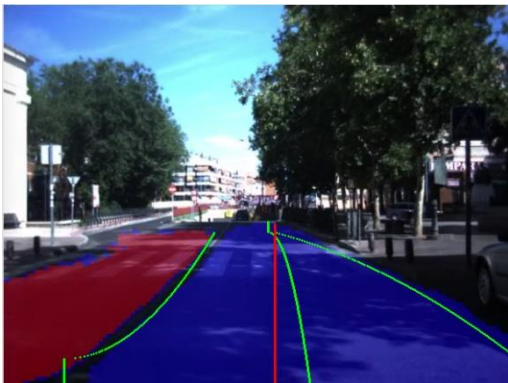


Fig. 33. Resultado imagen 4 modelo parabólico

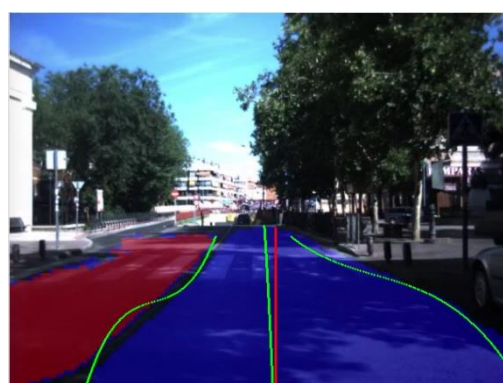


Fig. 34. Resultado imagen 4 modelo clotoidal

Imagen 5

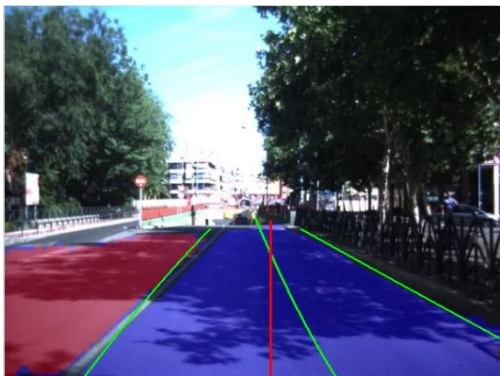


Fig. 35. Resultado imagen 5 modelo parabólico

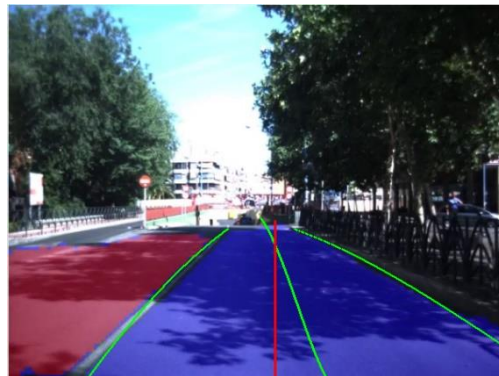


Fig. 36. Resultado imagen 5 modelo clotoidal

Imagen 6

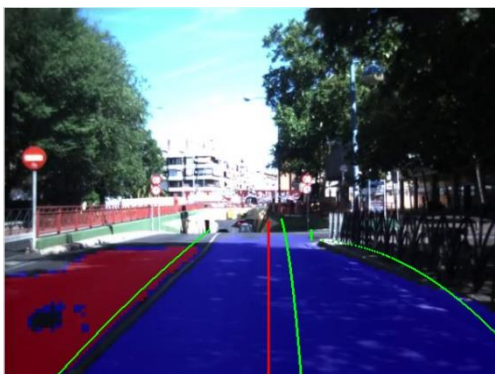


Fig. 37. Resultado imagen 6 modelo parabólico

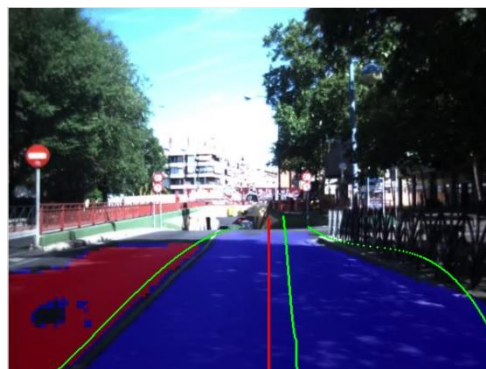


Fig. 38. Resultado imagen 6 modelo clotoidal

Los resultados de estas imágenes muestran como ambos modelos responden correctamente a la trayectoria del carril en los metros inmediatos al vehículo. Por el contrario, en los metros más cercanos al horizonte de la carretera, el modelo clotoidal se adapta de mejor manera, es decir, este tipo de curva le permite adaptarse a los cambios de trayectoria y al ruido recibido por parte de la detección del carril. Un ejemplo claro son las imágenes 1 y 2, en las que el modelo clotoidal (Fig. 28. y Fig. 30.) es capaz de variar su curvatura para adecuarse a los últimos metros percibidos del carril.

En la imagen 3, también se ve como el modelo parabólico (Fig. 31.) no prevé la pequeña curva que viene a continuación. Por el contrario, en la imagen 4, inmediatamente después, con la mejora de la percepción del carril, sí que consigue predecirla (Fig. 33.). Aun así, comparando estos resultados con los del modelo clotoidal (Fig. 32. y Fig. 34.), el modelo parabólico, al igual que en las imágenes anteriores, ofrece una desviación angular de

mayor grado y, por lo tanto, un cambio más brusco de trayectoria, lo cual será un punto a evitar en el control del automóvil.

Sin embargo, en la imagen 5, en la que se ofrece una delimitación mejor definida del carril, se observan resultados similares para ambos modelos. En cuanto al desplazamiento lateral, ambos coinciden en que el automóvil se encuentra un 12% desplazado a la izquierda. La desviación angular es similar, pero el modelo parabólico (Fig. 35.) sigue mostrando un ángulo de desviación de grado mayor al modelo clotoïdial (Fig. 36.), $-25,7^\circ$ contra $-19,5^\circ$.

Finalmente, observando la imagen 6, ambos modelos se aprecian desplazados ligeramente a la derecha, originado por la distorsión que provoca el desvío de una calle hacia dicho lado

A continuación, en la Tabla I, se muestran los resultados numéricos de las variables obtenidas para la coordenada de la imagen $y = 350$, es decir, unos pocos metros por delante del vehículo.

<p>Tabla I</p> <p>RESULTADOS</p>					
Imagen	Modelo de Carretera	Desviación angular	Distancia lateral Izq.	Distancia lateral Der.	Dist.lat. (%)
1	Parabólico	-27.154°	102,416	171,326	-25,16
	Clotoïdial	$-17,15^\circ$	119,363	170,959	-17.76
2	Parabólico	$-18,3404^\circ$	112,353	174,326	-21,60
	Clotoïdial	-13.637°	136,371	175,231	-12,46
3	Parabólico	$-1,6581^\circ$	140,754	207,354	-19,12
	Clotoïdial	$-12,3053^\circ$	154,309	226,695	-18,98
4	Parabólico	$-11,7995^\circ$	132,234	200,854	-20,6
	Clotoïdial	$-2,94641^\circ$	172,628	221,708	-12,44
5	Parabólico	$-25,6744^\circ$	128,334	163,926	-12,16
	Clotoïdial	$-19,5501^\circ$	133,225	172,01	-12,70
6	Parabólico	$-7,06673^\circ$	147,792	212,631	-17,98
	Clotoïdial	$-4,5054^\circ$	148,967	236,103	-22,62

4. PLANIFICACIÓN

En este epígrafe se detallan las fases por las cuales ha pasado el proyecto y el tiempo implementado en cada una de ellas.

Este Trabajo de Fin de Grado se divide en varias grandes fases:

- Revisión del estado del arte, estudiar el problema planteado y buscar posibles soluciones.
- Aprendizaje de los entornos de simulación.
- Desarrollo del software.
- Análisis de los resultados.
- Escritura de la memoria.

El proyecto comienza a principios de 2019, compaginando prácticas en empresa, dedicando una media de 3 horas diarias los días lectivos y entre 4 y 6 horas los días festivos y fines de semana.

A continuación, mediante un diagrama de Gantt, se puede ver de manera detallada la planificación del proyecto.

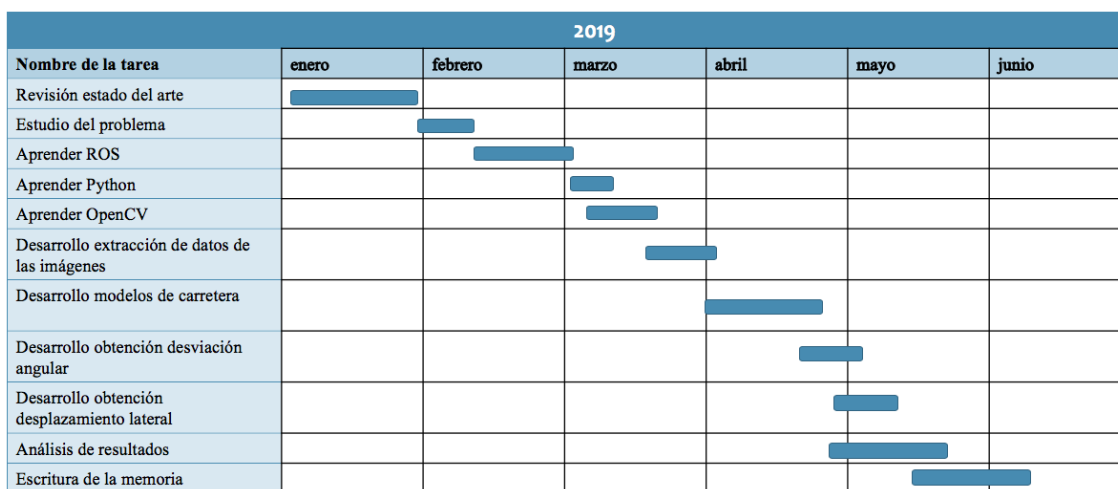


Fig. 39. Diagrama de Gantt de la planificación

5. MARCO REGULADOR

En este apartado se expondrá el marco legal y regulador actual en relación con la conducción autónoma, principalmente, en España y Europa.

En el Código de Tráfico y Seguridad Vial expedido por la Agencia Estatal Boletín Oficial del Estado, se recoge que “el conductor debe estar en todo momento en condiciones de controlar su vehículo” [49] y que “el conductor de vehículos a motor es responsable, en virtud del riesgo creado por la conducción de estos, de los daños causados a las personas o en los bienes con motivo de la circulación. En el caso de daños a las personas, de esta responsabilidad sólo quedará exonerado cuando pruebe que los daños fueron debidos a la culpa exclusiva del perjudicado o a fuerza mayor extraña a la conducción o al funcionamiento del vehículo; no se considerarán casos de fuerza mayor los defectos del vehículo ni la rotura o fallo de alguna de sus piezas o mecanismos” [50]. Estos artículos dejan en manos del conductor toda responsabilidad y control del vehículo, suponiendo inviable una conducción autónoma completa en la que el conductor no tendría que mostrarse dispuesto a interferir en las decisiones y trayectorias tomadas por el vehículo en todo momento.

Sin embargo, en 2015, La Dirección General de Tráfico (DGT) expidió una instrucción con motivo de la “autorización de pruebas o ensayos de investigación realizados con vehículos de conducción automatizada en vías abiertas al tráfico en general” [22], en la cual se define el concepto de vehículo autónomo, sus diferentes niveles de automatización y los requisitos y métodos necesarios para solicitar este permiso. Otra instrucción, expedida en 2016 por este mismo organismo estatal, recoge las condiciones de uso de los sistemas de estacionamiento asistido [51] y, a finales de 2017, se aprobó en el Congreso de los Diputados una Proposición no de Ley en la que se insta al Gobierno a renovar el marco legal de los vehículos autónomos e impulsar su desarrollo [52].

En el ámbito europeo, Alemania encabeza el desarrollo e inclusión de los vehículos autónomos en sus carreteras, desde 2017, cuando aprobó una Ley para regular el tráfico de vehículos autónomos [53]. Esta ley obliga a la existencia de un conductor que pueda tomar el control del automóvil en cualquier momento y a la incorporación de una caja negra que pueda esclarecer los hechos en caso de accidente. De este modo, se garantiza el conocer el responsable en caso de siniestro: la compañía del vehículo si el vehículo está en modo autónomo o el conductor.

Desde el Parlamento Europeo se aprobó en enero de 2019 una resolución sobre la conducción autónoma [54] en la que se subraya a la Comisión y a los Estados miembros la necesidad de adoptar una nueva legislación, garantizar unas infraestructuras, definir

unas directrices éticas claras y eliminar las limitaciones de responsabilidad en relación con la naturaleza y alcance de la responsabilidad civil con arreglo a las normas de Derecho civil sobre la robótica [55]. Al igual que será necesario seguir modificando ciertas enmiendas de la Convención de Viena, donde se firmaron la Convención sobre la Circulación Vial y a Convención sobre Señalización Vial en 1968, como ya hizo la Comisión Económica de Naciones Unidas para Europa (UNECE) en 2016 [56].

6. ESTUDIO SOCIO-ECONÓMICO

En este apartado se realiza un estudio socio-económico de lo que implicaría el desarrollo del proyecto conjunto llevado a cabo por el Laboratorio de Sistemas de la Universidad Carlos III de Madrid, es decir, el impacto a nivel social y económico del desarrollo y lanzamiento al mercado de un vehículo autónomo. También se detalla el presupuesto, en concreto, de este Trabajo de Fin de Grado.

6.1 Impacto social y económico

El lanzamiento y normalización de los vehículos autónomos en el mercado supondría una mejora en la calidad de vida humana, ya que solucionaría muchos de los problemas actuales gracias a la eliminación del error humano y la visión y planificación conjunta llevada a cabo por una red de transportes totalmente automatizada y adaptada a todo tipo de personas y necesidades.

Al poder calcular y realizar el trayecto óptimo sin errores y a una velocidad constante y adecuada a la vía, el ahorro de tiempo, energía y emisiones sería considerable. Como ya se ha visto en la introducción, esto supondría un ahorro económico tanto para el individuo como para el estado.

6.2 Presupuesto

El presupuesto estimado para llevar a cabo este proyecto se puede dividir entre los costes de personal y los costes materiales.

En cuanto al primer gasto, hay que tener en cuenta el trabajo realizado por un estudiante de Grado y su tutor, Ingeniero Telemático de Sistemas Inteligentes, durante seis meses.

- Trabajo realizado por el alumno durante seis meses: 20 horas semanales.

- Asesoramiento, reuniones y participación por parte del tutor durante el proyecto: 1h semanal.

Teniendo en cuenta el coste salarial de cada uno de los miembros del proyecto, se puede realizar una estimación del valor salarial de personal que supondría la realización de este Trabajo de Fin de Grado, detallado a continuación en la Tabla II.

Tabla II			
Coste de Personal			
Categorías	Salario/hora	Horas trabajadas	Total
Tutor	50€	24	1200€
Estudiante	9€	480	4320€
Coste Total			5520€

En relación a los costes materiales, se ha utilizado un ordenador portátil Acer Aspire F5-572G-517R, valorado en 569€. Todos los softwares utilizados, Ubuntu 16.04, ROS Kinetic y OpenCV, son de acceso libre y han supuesto coste cero.

Para la obtención de las imágenes se ha utilizado la cámara Basler con un sensor CMOS de 3.2MP y una óptica de 3.5 mm, con un FOV frontal de 90°. Esta cámara junto con la óptica tiene un precio de 1100€.

Teniendo en cuenta el precio, el periodo de amortización y el tiempo utilizado, el coste real de estos materiales en el proyecto han sido los siguientes:

Tabla III				
Costes materiales				
Producto	Precio	Tiempo de amortización (meses)	Tiempo de uso (meses)	Coste
Ordenador	569€	48	6	71,13€
Cámara	1100€	12	1	91,7€
Total				162€

Para calcular los costes materiales del proyecto se ha utilizado la siguiente formula:

$$Coste = \frac{Precio}{Tiempo\ de\ Amortización} \times Tiempo\ de\ Uso \quad (54)$$

Por lo tanto, el total de gastos de este proyecto ascenderían a 5682,83€.

A continuación, se muestran los gastos totales detallados en la tabla IV.

Tabla IV	
Resumen del Presupuesto	
Coste de personal	
• Estudiante	4320€
• Tutor	1440€
Coste del Hardware	
• Ordenador	71,13€
• Cámara Basler	91,7€
Costes de Software	
• Ubuntu 16.04	0€
• ROS Kinetic	0€
• OpenCV	0€
Total	5682,83€

7. CONCLUSIÓN

Tras la validación del sistema basada en unos resultados cualitativos, se puede observar en los resultados llevados a cabo, la posibilidad de implementar el sistema en tiempo real junto con el control lateral del vehículo, siendo este el principal objetivo del proyecto.

En cuanto al análisis de los modelos de carretera, ambos se muestran válidos y no se aprecian grandes diferencias entre ellos. Aun así, el modelo clotoidal ofrece unas prestaciones superiores al modelo parabólico, ya que cuenta con una variable más en el análisis y extracción de datos, lo que permite adaptarse mejor a la curvatura del carril. Esta diferencia es más notable cuando la detección del carril no ofrece una delimitación tan definida.

Ambos modelos ofrecen mayores garantías con el aumento de la precisión en la detección del carril, por lo que supone uno de los aspectos más importantes del sistema.

En aquellas imágenes en la que se encuentra mejor definido el carril, se puede observar unos resultados en ambos modelos similares. Sin embargo, el modelo parabólico tiende a ofrecer una mayor desviación angular. En el desplazamiento lateral no se aprecian grandes diferencias ni una tendencia visible entre ellos.

En cuanto a la implementación del sistema en un entorno real, ya que los resultados obtenidos son solidarios al control lateral del vehículo y, por ende, responsable en gran medida de la seguridad tanto del automóvil como de sus ocupantes y entorno, se debería llevar a cabo una validación completa, al ser posible, de manera cuantitativa y obteniendo el error real.

7.1 Líneas de trabajo futuro

Como se ha podido apreciar en el proyecto, uno de los aspectos que traería una mayor mejora en los resultados, sería la optimización de la detección del carril.

Para ello, y en relación con la extracción de datos de la carretera, también se podría utilizar un cambio de perspectivas en la imagen, usando una perspectiva inversa de mapeado [30]. De esta manera se eliminaría la distorsión de la imagen obtenida y se trabajaría con una curva plana. Es decir, se obtendría una imagen con distancias reales.

De igual modo, se podría usar el LiDAR calibrado con la cámara para hallar las coordenadas reales de cada píxel, obteniendo las distancias reales.

Una vez obtenidas estas distancias, se podría calcular mediante una integral la longitud (L) de la trayectoria curva seguida por la carretera.

$$L = \int_{x=a}^{x=b} \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dy$$

Teniendo en cuenta el tiempo empleado en calcular la desviación angular y las distancias laterales, en corregir el control lateral y la velocidad del vehículo, se podría mejorar y predecir la distancia del vehículo al punto de referencia utilizado para obtener las variables.

De este modo, el control lateral del vehículo se podría adaptar y prever su posición en la carretera dependiendo de la velocidad del automóvil.

Otro aspecto a trabajar sería la creación de una base de datos más amplia en la que se tuviera conocimiento de las medidas reales de la carretera a la hora de comparar y verificar los resultados del sistema.

BIBLIOGRAFIA

- [1] «Historia del automóvil,» Wikipedia, [En línea]. Available: https://es.wikipedia.org/wiki/Historia_del_autom%C3%B3vil. [Último acceso: 01 06 2019].
- [2] C. Engelking, «The ‘Driverless’ Car Era Began More Than 90 Years Ago,» Discover Magazine, 13 12 17. [En línea]. Available: <http://blogs.discovermagazine.com/d-brief/2017/12/13/driverless-car-houdina-houdini/#.XNRQy44zbIU>. [Último acceso: 01 06 2019].
- [3] P. G. Bejerano, «Futurama o cómo se veía el futuro en 1939,» eldiario.es, 07 05 2014. [En línea]. Available: https://www.eldiario.es/turing/futurama-1939_0_256875095.html. [Último acceso: 06 06 2019].
- [4] J. Delcker, «The man who invented the self-driving car (in 1986),» Politico, 19 07 2018. [En línea]. Available: <https://www.politico.eu/article/delf-driving-car-born-1986-ernst-dickmanns-mercedes/>. [Último acceso: 06 06 2019].
- [5] «The Grand Challenge,» DARPA, 2005. [En línea]. Available: <https://www.darpa.mil/about-us/timeline/-grand-challenge-for-autonomous-vehicles>. [Último acceso: 06 06 2019].
- [6] «Roborace,» [En línea]. Available: <https://roborace.com/>. [Último acceso: 06 06 2019].
- [7] «SPEEDING TO HOCKENHEIM: HOW STUDENTS ARE BUILDING AN AUTONOMOUS RACE CAR,» 2025 AD, 13 06 2018. [En línea]. Available: <https://www.2025ad.com/latest/formula-student-driverless/>. [Último acceso: 06 06 2019].
- [8] «Uber pone en circulación sus primeros coches 'autónomos' en Pittsburgh,» El periódico de la energía, 14 09 2016. [En línea]. Available: <https://elperiodicodelaenergia.com/uber-pone-en-circulacion-sus-primeros-coches-autonomos-en-pittsburgh/>. [Último acceso: 08 06 2019].
- [9] «Google estrena sus taxis autónomos: así es viajar en uno de ellos,» El Confidencial, 17 12 2018. [En línea]. Available: https://www.elconfidencial.com/multimedia/album/tecnologia/2018-12-09/viajar-waymo-coche-autonomo-google_1692842#0. [Último acceso: 08 06 2019].
- [10] «Google levanta la primera fábrica de coches autónomos del mundo,» Economía digital, 25 01 2019. [En línea]. Available: https://www.economiadigital.es/tecnologia-y-tendencias/google-levanta-la-primera-fabrica-de-coches-autonomos-del-mundo_601799_102.html. [Último acceso: 06 08 2019].
- [11] «El futuro de la conducción,» Tesla, [En línea]. Available: https://www.tesla.com/es_ES/autopilot. [Último acceso: 08 06 2019].
- [12] «Las 10 principales causas de defunción,» OMS, 24 05 2018. [En línea]. Available: <https://www.who.int/es/news-room/fact-sheets/detail/the-top-10-causes-of-death>. [Último acceso: 01 06 2019].

- [13] A. Gutiérrez, «Los accidentes de tráfico, principal causa de muerte en jóvenes,» DGT, Dirección General de Tráfico, 17 12 2018. [En línea]. Available: <http://revista.dgt.es/es/noticias/internacional/2018/1218oms-informe-mundial-accidentes-trafico.shtml#.XNfJOI4zbIU>. [Último acceso: 01 06 2019].
- [14] «Presentación balance de accidentes en 2018,» DGT, Dirección General de Tráfico, 03 01 2019. [En línea]. Available: <http://revista.dgt.es/es/noticias/nacional/2019/01ENERO/0103-Presentacion-balance-accidentes-2018.shtml#.XNfPiY4zbIU>. [Último acceso: 01 06 2019].
- [15] «La velocidad excesiva o inadecuada sigue siendo una de las principales causas de los accidentes de tráfico,» DGT, Dirección General de Tráfico, 01 04 2019. [En línea]. Available: <http://www.dgt.es/es/prensa/notas-de-prensa/2019/La-velocidad-excesiva-o-inadecuada-sigue-siendo-una-de-las-principales-causas-de-los-accidentes-de-trafico.shtml>. [Último acceso: 01 06 2019].
- [16] «Congestion Costs U.K. Nearly £8 Billion in 2018,» INRIX, 12 02 2019. [En línea]. Available: <http://inrix.com/press-releases/scorecard-2018-uk/>. [Último acceso: 01 06 2019].
- [17] «INRIX: Congestion Costs Each American 97 hours, \$1,348 A Year,» INRIX, 11 02 2019. [En línea]. Available: <http://inrix.com/press-releases/scorecard-2018-us/>. [Último acceso: 01 06 2019].
- [18] «Interactive Ranking & City Dashboards,» INRIX, 2018. [En línea]. Available: <http://inrix.com/scorecard/>. [Último acceso: 01 06 2019].
- [19] «Emisiones de CO2 de los coches: hechos y cifras (infografía),» Parlamento Europeo, 22 03 2019. [En línea]. Available: <http://www.europarl.europa.eu/news/es/headlines/society/20190313STO31218/emisiones-de-co2-de-los-coches-hechos-y-cifras-infografia>. [Último acceso: 01 06 2019].
- [20] «Europa acuerda reducir un 37,5% las emisiones de CO2 en coches para 2030» EFE Verde, 18 12 2018. [En línea]. Available: <https://www.efeverde.com/noticias/reducir-375-emisiones-co2-coches-eurocamara-consejo/>. [Último acceso: 01 06 2019].
- [21] «Restricciones de tráfico en el centro de Madrid,» OCU, 23 04 2019. [En línea]. Available: <https://www.ocu.org/coches/coches/noticias/restricciones-trafico-madrid>. [Último acceso: 01 06 2019].
- [22] M. D. I. -. D. G. D. TRÁFICO, «INSTRUCCIÓN 15/V-113,» Madrid, SUBDIRECCIÓN GENERAL DE GESTIÓN DE LA MOVILIDAD, 2015.
- [23] «Moral Machine» MIT, Massachusetts Institute of Technology, [En línea]. Available: <http://moralmachine.mit.edu/hl/es>. [Último acceso: 01 06 2019].
- [24] R. J. Cano, «Primer atropello mortal de un coche sin conductor,» El País, 20 03 2018. [En línea]. Available: https://elpais.com/tecnologia/2018/03/19/actualidad/1521479089_032894.html. [Último acceso: 08 06 2019].
- [25] D. Hernández, «Estudio, análisis y clasificación de los sistemas ADAS incorporados en los vehículos,» Valladolid: Universidad de Valladolid, 2018.
- [26] «Lane Keeping Assist Systems,» VDA, Verband der Automobilindustrie, [En línea]. Available: <https://www.vda.de/en/topics/safety-and-standards/lkas/lane-keeping-assist-systems.html>. [Último acceso: 08 06 2019].
- [27] «Lane Keeping Assist» Continental, [En línea]. Available: <https://www.continental-automotive.com/en-gl/Passenger-Cars/Chassis-Safety/Software-Functions/Cruising/Lane-Keeping-Assist>. [Último acceso: 08 06 2019].
- [28] Y.-W. S. a. R. (. Rajkumar, «Tracking and Estimation of Ego-Vehicle's State for

- Lateral Localization» de IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 2014.
- [29] J.-H. W. Y.-F. S. Jing-Fu Liu, «Development of an Interactive Lane Keeping Control System for Vehicle» Automotive Research and Testing Center (ARTC), Changhua County 505, Taiwan, R.O.C.
 - [30] J. J. a. A. Kim, «Adaptive Inverse Perspective Mapping for Lane Map Generation with SLAM» de 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Xian, China, 2016.
 - [31] Y. G. S. K. M. I. Jiali Bao, «Vehicle positioning with the integration of scene understanding and 3D map in urban environment» de IEEE Intelligent Vehicles Symposium (IV), Redondo Beach, CA, USA, 2017.
 - [32] S.-H. L. a. C. C. Chung, «Robust Multirate On-Road Vehicle Localization for Autonomous Highway Driving Vehicles» TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, vol. 25, nº 2, 2017.
 - [33] K. Z. a. N. Geiß, «Vehicle Ego-Localization in Autonomous Lane-Keeping Evasive Maneuvers» de IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd., Aschaffenburg, Germany, 2016.
 - [34] L. y. J. O. y. S. I. Sánchez, «Filtro de Kalman extendido y filtro de partículas Kalman extendido para problemas de estimación No Lineal» Revista INGENIERÍA UC, vol. 20, nº 1, pp. 7-16, 2013.
 - [35] F. T. y. M. C. Cerrajero, «Cálculo, Material de Clase,» Universidad Carlos III de Madrid, [En línea]. Available: <http://ocw.uc3m.es/matematicas/calculo/material-de-clase-1/capitulo-5.pdf>. [Último acceso: 08 06 2016].
 - [36] M. O. R. Nave, «Series de Taylor,» Hyperphysics, [En línea]. Available: <http://hyperphysics.phy-astr.gsu.edu/hbasees/tayser.html>. [Último acceso: 08 06 2019].
 - [37] Y. G. S. K. Jiali Bao, «Vehicle positioning with the integration of scene understanding and 3D map in urban environment» de IEEE Intelligent Vehicles Symposium (IV), Redondo Beach, CA, USA, 2017.
 - [38] Y. Narita, S. Katahara y M. Aoki, «Lateral position detection using side looking line sensor cameras» de IEEE IV2003 Intelligent Vehicles Symposium, Columbus, OH, USA, 2003.
 - [39] S.-W. K. a. S.-W. S. Soomok Lee, «Accurate Ego-Lane Recognition utilizing Multiple Road Characteristics in a Bayesian Network Framework» de 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 2015.
 - [40] S. O. S. K. a. J. M. Z. Florian Kuhnt, «Lane-precise Localization of Intelligent Vehicles Using the Surrounding Object Constellation» de IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 2016.
 - [41] D. C. R. I. I. P. M. A. S. D. G. S. A. L. Ballardini, «Ego-Lane Estimation by Modeling Lanes and Sensor Failures» de IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017.
 - [42] «ROS,» [En línea]. Available: <http://wiki.ros.org/es>. [Último acceso: 08 06 2019].
 - [43] «OpenCV,» [En línea]. Available: <https://opencv.org/>.
 - [44] F. Pizzati y F. García, «Enhanced free space detection in multiple lanes based on single CNN with scene identification» de IEEE Intelligent Vehicles Symposium (IV).
 - [45] «Parábola (matemática),» Wikipedia, [En línea]. Available: [https://es.wikipedia.org/wiki/Par%C3%A1bola_\(matem%C3%A1tica\)](https://es.wikipedia.org/wiki/Par%C3%A1bola_(matem%C3%A1tica)). [Último acceso: 01 06 2019].

- [46] E. C. J. M. Luís Blanch, «Una aproximación a la curva de transición,» *Modelling in Science Education and Learning*, vol. 6, nº 9, 2013.
- [47] «Clotoide,» Treccani, [En línea]. Available: http://www.treccani.it/enciclopedia/clotoide_%28Enciclopedia-della-Matematica%29/. [Último acceso: 09 06 2019].
- [48] «Cómo resolver una ecuación cúbica» WikiHow, [En línea]. Available: <https://es.wikihow.com/resolver-una-ecuaci%C3%B3n-c%C3%BAbica>. [Último acceso: 01 06 2019].
- [49] M. D. I. -. D. G. D. TRÁFICO, «Texto refundido de la Ley sobre Tráfico, Circulación de Vehículos a Motor y Seguridad Vial,» de Código de Tráfico y Seguridad Vial, Agencia Estatal Boletín Oficial del Estado, 2019, p. 9.
- [50] M. D. I. -. D. G. D. TRÁFICO, «TEXTO REFUNDIDO DE LA LEY SOBRE RESPONSABILIDAD CIVIL» de Código de Tráfico y Seguridad Vial, Agencia Estatal Boletín Oficial del Estado, 2019, p. 493.
- [51] M. D. I. -. D. G. D. TRÁFICO, «INSTRUCCIÓN 16 TV/89,» Madrid, Subdirección General de Tráfico, 2016.
- [52] P. d. C. d. l. Diputados, «Impulso y desarrollo del vehículo autónomo,» de Texto de la Proposición no de Ley (núm. expte. 162/000451), 10 de octubre de 2017.
- [53] T. Escritt, «Germany adopts self-driving vehicles law,» Reuters, 12 05 2017. [En línea]. Available: <https://www.reuters.com/article/us-germany-autos-self-driving-idUSKBN1881HY>. [Último acceso: 02 06 2019].
- [54] P. Europeo, «Sobre la conducción autónoma en los transportes europeos (2018/2089(INI)),» de Resolución del Parlamento Europeo, Estrasburgo, 15 de enero de 2019.
- [55] A. C. A. C. E. Y. S. E. Y. A. C. D. L. R. COMUNICACIÓN DE LA COMISIÓN AL PARLAMENTO EUROPEO, «En ruta hacia la movilidad automatizada: estrategia de la UE para la movilidad del futuro,» Bruselas, 2018.
- [56] C. E. P. E. -. D. d. Transporte, «Convención sobre la Circulación Vial de 1968 y Acuerdo Europeo que complementa la Convención,» Naciones Unidas, 2016.

ANEXO. ESPECIFICACIONES TÉCNICAS DE LA CÁMARA BASLER

General Information

Order Number	107215
Product Line	ace U

Sensor

Sensor Vendor	Sony
Sensor	IMX265
Shutter	Global Shutter
Max. Image Circle	1/1.8"
Sensor Type	CMOS
Sensor Size	7.1 mm x 5.3 mm
Resolution (HxV)	2048 px x 1536 px
Resolution	3 MP
Pixel Size (H x V)	3.45 µm x 3.45 µm
Frame Rate	36 fps
Mono/Color	Color

Camera Data

Interface	GigE
Pixel Bit Depth	10 or 12 bits
Synchronization	<ul style="list-style-type: none">▪ software trigger▪ hardware trigger▪ free-run
Exposure Control	<ul style="list-style-type: none">▪ programmable via the camera API▪ hardware trigger
Digital Input	1
Digital Output	1
General Purpose I/O	1
Power Requirements	<ul style="list-style-type: none">▪ PoE or 12-24 VDC
Power Consumption (typical)	2.7 W
Power Consumption PoE	3.3 W

Housing

Housing Type	Box
Housing Size (L x W x H)	42 mm x 29 mm x 29 mm
Lens Mount	<ul style="list-style-type: none">▪ C-mount
Housing Temperature	0 - 50°C
Weight (typical)	90g

